

FreeBSD kernel netgraph code Reference Manual

Generated by Doxygen 1.4.7

Sat Feb 24 18:04:45 2007

Contents

1	FreeBSD kernel netgraph code Main Page	1
2	FreeBSD kernel netgraph code Directory Hierarchy	3
2.1	FreeBSD kernel netgraph code Directories	3
3	FreeBSD kernel netgraph code Data Structure Index	5
3.1	FreeBSD kernel netgraph code Data Structures	5
4	FreeBSD kernel netgraph code File Index	13
4.1	FreeBSD kernel netgraph code File List	13
5	FreeBSD kernel netgraph code Directory Documentation	17
5.1	/usr/src/sys/netgraph/atm/ Directory Reference	17
5.2	/usr/src/sys/netgraph/atm/atmpif/ Directory Reference	18
5.3	/usr/src/sys/netgraph/bluetooth/ Directory Reference	19
5.4	/usr/src/sys/netgraph/bluetooth/drivers/bt3c/ Directory Reference	20
5.5	/usr/src/sys/netgraph/atm/ccatm/ Directory Reference	21
5.6	/usr/src/sys/netgraph/bluetooth/common/ Directory Reference	22
5.7	/usr/src/sys/netgraph/bluetooth/drivers/ Directory Reference	23
5.8	/usr/src/sys/netgraph/bluetooth/drivers/h4/ Directory Reference	24
5.9	/usr/src/sys/netgraph/bluetooth/hci/ Directory Reference	25
5.10	/usr/src/sys/netgraph/bluetooth/include/ Directory Reference	26
5.11	/usr/src/sys/netgraph/bluetooth/l2cap/ Directory Reference	27
5.12	/usr/src/sys/netgraph/netflow/ Directory Reference	28
5.13	/usr/src/sys/netgraph/ Directory Reference	29
5.14	/usr/src/sys/netgraph/bluetooth/socket/ Directory Reference	32
5.15	/usr/src/ Directory Reference	33
5.16	/usr/src/sys/netgraph/atm/sscfu/ Directory Reference	34
5.17	/usr/src/sys/netgraph/atm/sscop/ Directory Reference	35
5.18	/usr/src/sys/ Directory Reference	36

5.19	/usr/src/sys/netgraph/bluetooth/drivers/ubt/ Directory Reference	37
5.20	/usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ Directory Reference	38
5.21	/usr/src/sys/netgraph/atm/uni/ Directory Reference	39
5.22	/usr/ Directory Reference	40
6	FreeBSD kernel netgraph code Data Structure Documentation	41
6.1	__attribute__ Struct Reference	41
6.2	bt3c_softc Struct Reference	54
6.3	ccatm_op Struct Reference	58
6.4	cchook Struct Reference	59
6.5	ccnode Struct Reference	61
6.6	cisco_header Struct Reference	63
6.7	cisco_packet Struct Reference	64
6.8	cisco_priv Struct Reference	66
6.9	ctxinfo Struct Reference	69
6.10	datatag Struct Reference	71
6.11	ETF Struct Reference	72
6.12	ETF_hookinfo Struct Reference	74
6.13	filter Struct Reference	75
6.14	flow_entry Struct Reference	76
6.15	flow_entry_data Struct Reference	77
6.16	flow_hash_entry Struct Reference	79
6.17	flow_manager Struct Reference	80
6.18	flow_rec Struct Reference	81
6.19	firmrel_softc Struct Reference	83
6.20	greheader Struct Reference	85
6.21	hookinfo Struct Reference	87
6.22	hooklist Struct Reference	88
6.23	hpriv_p Struct Reference	89
6.24	hva_stats_aal5 Struct Reference	90
6.25	hva_stats_atm Struct Reference	92
6.26	hva_stats_ng Struct Reference	93
6.27	iffam Struct Reference	95
6.28	int16_temp Struct Reference	96
6.29	int32_temp Struct Reference	97
6.30	int64_temp Struct Reference	98
6.31	l2tp_seq Struct Reference	99

6.32 linkinfo Struct Reference	102
6.33 namelist Struct Reference	103
6.34 netflow Struct Reference	104
6.35 netflow_v1_header Struct Reference	106
6.36 netflow_v1_record Struct Reference	107
6.37 netflow_v5_export_dgram Struct Reference	110
6.38 netflow_v5_header Struct Reference	111
6.39 netflow_v5_record Struct Reference	113
6.40 ng_async_cfg Struct Reference	116
6.41 ng_async_private Struct Reference	117
6.42 ng_async_stat Struct Reference	120
6.43 ng_atmllc_priv Struct Reference	122
6.44 ng_atmpif_link_status Struct Reference	124
6.45 ng_bpf_hookinfo Struct Reference	125
6.46 ng_bpf_hookprog Struct Reference	127
6.47 ng_bpf_hookstat Struct Reference	128
6.48 ng_bridge_config Struct Reference	130
6.49 ng_bridge_hent Struct Reference	131
6.50 ng_bridge_host Struct Reference	132
6.51 ng_bridge_host_ary Struct Reference	133
6.52 ng_bridge_link Struct Reference	134
6.53 ng_bridge_link_stats Struct Reference	136
6.54 ng_bridge_private Struct Reference	139
6.55 ng_bt3c_firmware_block_ep Struct Reference	141
6.56 ng_bt3c_node_qlen_ep Struct Reference	142
6.57 ng_bt3c_node_stat_ep Struct Reference	143
6.58 ng_bt_itemq Struct Reference	144
6.59 ng_bt_mbufq Struct Reference	146
6.60 ng_btsocket_hci_raw_con_list Struct Reference	147
6.61 ng_btsocket_hci_raw_filter Struct Reference	148
6.62 ng_btsocket_hci_raw_node_bdaddr Struct Reference	149
6.63 ng_btsocket_hci_raw_node_buffer Struct Reference	150
6.64 ng_btsocket_hci_raw_node_debug Struct Reference	151
6.65 ng_btsocket_hci_raw_node_features Struct Reference	152
6.66 ng_btsocket_hci_raw_node_link_policy_mask Struct Reference	153
6.67 ng_btsocket_hci_raw_node_list_names Struct Reference	154

6.68	ng_btsocket_hci_raw_node_neighbor_cache Struct Reference	155
6.69	ng_btsocket_hci_raw_node_packet_mask Struct Reference	156
6.70	ng_btsocket_hci_raw_node_role_switch Struct Reference	157
6.71	ng_btsocket_hci_raw_node_stat Struct Reference	158
6.72	ng_btsocket_hci_raw_node_state Struct Reference	159
6.73	ng_btsocket_hci_raw_pcb Struct Reference	160
6.74	ng_btsocket_hci_raw_sec_filter Struct Reference	162
6.75	ng_btsocket_l2cap_pcb Struct Reference	163
6.76	ng_btsocket_l2cap_raw_auto_discon_timo Struct Reference	168
6.77	ng_btsocket_l2cap_raw_chan_list Struct Reference	169
6.78	ng_btsocket_l2cap_raw_con_list Struct Reference	170
6.79	ng_btsocket_l2cap_raw_get_info Struct Reference	171
6.80	ng_btsocket_l2cap_raw_node_debug Struct Reference	172
6.81	ng_btsocket_l2cap_raw_node_flags Struct Reference	173
6.82	ng_btsocket_l2cap_raw_pcb Struct Reference	174
6.83	ng_btsocket_l2cap_raw_ping Struct Reference	177
6.84	ng_btsocket_l2cap_rtentry Struct Reference	178
6.85	ng_btsocket_rfcomm_fc_info Struct Reference	180
6.86	ng_btsocket_rfcomm_pcb Struct Reference	182
6.87	ng_btsocket_rfcomm_session Struct Reference	186
6.88	ng_cisco_ipaddr Struct Reference	189
6.89	ng_cisco_stats Struct Reference	190
6.90	ng_cmdlist Struct Reference	191
6.91	ng_deflate_config Struct Reference	193
6.92	ng_deflate_private Struct Reference	194
6.93	ng_deflate_stats Struct Reference	196
6.94	ng_eiface_private Struct Reference	197
6.95	ng_etffilter Struct Reference	199
6.96	ng_etfstat Struct Reference	200
6.97	ng_fec_bundle Struct Reference	201
6.98	ng_fec_ifname Struct Reference	202
6.99	ng_fec_portlist Struct Reference	203
6.100	ng_fec_private Struct Reference	204
6.101	ng_gif_demux_private Struct Reference	206
6.102	ng_h4_info Struct Reference	208
6.103	ng_h4_node_stat_ep Struct Reference	211

6.104ng_hci_lp_con_cfm_ep Struct Reference	212
6.105ng_hci_lp_con_ind_ep Struct Reference	213
6.106ng_hci_lp_con_req_ep Struct Reference	214
6.107ng_hci_lp_con_rsp_ep Struct Reference	215
6.108ng_hci_lp_discon_ind_ep Struct Reference	216
6.109ng_hci_lp_discon_req_ep Struct Reference	217
6.110ng_hci_lp_qos_cfm_ep Struct Reference	218
6.111ng_hci_lp_qos_ind_ep Struct Reference	219
6.112ng_hci_lp_qos_req_ep Struct Reference	220
6.113ng_hci_neighbor Struct Reference	222
6.114ng_hci_node_buffer_ep Struct Reference	224
6.115ng_hci_node_con_ep Struct Reference	226
6.116ng_hci_node_con_list_ep Struct Reference	228
6.117ng_hci_node_get_neighbor_cache_ep Struct Reference	229
6.118ng_hci_node_neighbor_cache_entry_ep Struct Reference	230
6.119ng_hci_node_stat_ep Struct Reference	231
6.120ng_hci_node_up_ep Struct Reference	233
6.121ng_hci_sync_con_queue_ep Struct Reference	234
6.122ng_hci_unit Struct Reference	235
6.123ng_hci_unit_buff Struct Reference	239
6.124ng_hci_unit_con Struct Reference	241
6.125ng_hole_hookinfo Struct Reference	244
6.126ng_hole_hookstat Struct Reference	245
6.127ng_hook Struct Reference	246
6.128ng_iface_private Struct Reference	249
6.129ng_ipfw_hook_priv Struct Reference	251
6.130ng_ipfw_tag Struct Reference	252
6.131ng_item Struct Reference	253
6.132ng_ksocket_accept Struct Reference	256
6.133ng_ksocket_alias Struct Reference	257
6.134ng_ksocket_private Struct Reference	258
6.135ng_ksocket_sockopt Struct Reference	260
6.136ng_l2cap Struct Reference	261
6.137ng_l2cap_cfg_opt_val_t Union Reference	264
6.138ng_l2cap_chan Struct Reference	265
6.139ng_l2cap_cmd Struct Reference	269

6.140ng_l2cap_cmd_rej_data_t Union Reference	271
6.141ng_l2cap_con Struct Reference	272
6.142ng_l2cap_info_rsp_data_t Union Reference	276
6.143ng_l2cap_l2ca_cfg_ind_ip Struct Reference	277
6.144ng_l2cap_l2ca_cfg_ip Struct Reference	278
6.145ng_l2cap_l2ca_cfg_op Struct Reference	279
6.146ng_l2cap_l2ca_cfg_rsp_ip Struct Reference	280
6.147ng_l2cap_l2ca_cfg_rsp_op Struct Reference	281
6.148ng_l2cap_l2ca_con_ind_ip Struct Reference	282
6.149ng_l2cap_l2ca_con_ip Struct Reference	283
6.150ng_l2cap_l2ca_con_op Struct Reference	284
6.151ng_l2cap_l2ca_con_rsp_ip Struct Reference	285
6.152ng_l2cap_l2ca_con_rsp_op Struct Reference	286
6.153ng_l2cap_l2ca_discon_ip Struct Reference	287
6.154ng_l2cap_l2ca_discon_op Struct Reference	288
6.155ng_l2cap_l2ca_enable_clt_ip Struct Reference	289
6.156ng_l2cap_l2ca_get_info_ip Struct Reference	290
6.157ng_l2cap_l2ca_get_info_op Struct Reference	291
6.158ng_l2cap_l2ca_grp_add_member_ip Struct Reference	292
6.159ng_l2cap_l2ca_grp_add_member_op Struct Reference	293
6.160ng_l2cap_l2ca_grp_close_ip Struct Reference	294
6.161ng_l2cap_l2ca_grp_create_ip Struct Reference	295
6.162ng_l2cap_l2ca_grp_create_op Struct Reference	296
6.163ng_l2cap_l2ca_grp_get_members_ip Struct Reference	297
6.164ng_l2cap_l2ca_grp_get_members_op Struct Reference	298
6.165ng_l2cap_l2ca_ping_ip Struct Reference	299
6.166ng_l2cap_l2ca_ping_op Struct Reference	300
6.167ng_l2cap_l2ca_qos_ind_ip Struct Reference	301
6.168ng_l2cap_l2ca_write_op Struct Reference	302
6.169ng_l2cap_node_chan_ep Struct Reference	303
6.170ng_l2cap_node_chan_list_ep Struct Reference	305
6.171ng_l2cap_node_con_ep Struct Reference	306
6.172ng_l2cap_node_con_list_ep Struct Reference	307
6.173ng_l2tp_config Struct Reference	308
6.174ng_l2tp_hook_private Struct Reference	310
6.175ng_l2tp_private Struct Reference	311

6.176ng_l2tp_seq_config Struct Reference	313
6.177ng_l2tp_sess_config Struct Reference	314
6.178ng_l2tp_session_stats Struct Reference	315
6.179ng_l2tp_stats Struct Reference	316
6.180ng_mesg Struct Reference	319
6.181ng_mesg::ng_msghdr Struct Reference	321
6.182ng_mppc_config Struct Reference	324
6.183ng_mppc_dir Struct Reference	325
6.184ng_mppc_private Struct Reference	327
6.185ng_nat_priv Struct Reference	328
6.186ng_netflow_iface Struct Reference	330
6.187ng_netflow_ifinfo Struct Reference	332
6.188ng_netflow_info Struct Reference	333
6.189ng_netflow_setdlt Struct Reference	335
6.190ng_netflow_setifindex Struct Reference	336
6.191ng_netflow_settimeouts Struct Reference	337
6.192ng_node Struct Reference	338
6.193ng_one2many_config Struct Reference	341
6.194ng_one2many_link Struct Reference	342
6.195ng_one2many_link_stats Struct Reference	343
6.196ng_one2many_private Struct Reference	344
6.197ng_parse_array_info Struct Reference	346
6.198ng_parse_fixedarray_info Struct Reference	347
6.199ng_parse_fixedstring_info Struct Reference	348
6.200ng_parse_struct_field Struct Reference	349
6.201ng_parse_type Struct Reference	350
6.202ng_ppp_bund_conf Struct Reference	352
6.203ng_ppp_frag Struct Reference	355
6.204ng_ppp_link Struct Reference	357
6.205ng_ppp_link_conf Struct Reference	359
6.206ng_ppp_link_stat Struct Reference	361
6.207ng_ppp_mp_state Struct Reference	363
6.208ng_ppp_node_conf Struct Reference	364
6.209ng_ppp_private Struct Reference	365
6.210ng_pptpgre_ackp Struct Reference	367
6.211ng_pptpgre_conf Struct Reference	369

6.212ng_pptgre_private Struct Reference	371
6.213ng_pptgre_stats Struct Reference	373
6.214ng_pred1_config Struct Reference	375
6.215ng_pred1_private Struct Reference	376
6.216ng_pred1_stats Struct Reference	378
6.217ng_queue Struct Reference	379
6.218ng_rfc1490_encap_t Struct Reference	381
6.219ng_rfc1490_private Struct Reference	382
6.220ng_source_stats Struct Reference	384
6.221ng_split_private Struct Reference	386
6.222ng_sppp_private Struct Reference	388
6.223ng_sscfu_getdefparam Struct Reference	390
6.224ng_sscop_setparam Struct Reference	391
6.225ng_sscop_setparam_resp Struct Reference	392
6.226ng_tag_hookin Struct Reference	393
6.227ng_tag_hookinfo Struct Reference	395
6.228ng_tag_hookout Struct Reference	397
6.229ng_tag_prio Struct Reference	398
6.230ng_tcpmss_config Struct Reference	399
6.231ng_tcpmss_hookstat Struct Reference	400
6.232ng_tee_hookstat Struct Reference	401
6.233ng_tee_stats Struct Reference	402
6.234ng_type Struct Reference	403
6.235ng_ubt_node_qlen_ep Struct Reference	406
6.236ng_ubt_node_stat_ep Struct Reference	407
6.237ng_UI_private Struct Reference	408
6.238ng_vatmpif_config Struct Reference	409
6.239ng_vatmpif_hook Struct Reference	410
6.240ng_vjc_private Struct Reference	412
6.241ng_vlan_filter Struct Reference	414
6.242ng_vlan_table Struct Reference	415
6.243ngatm_msg Struct Reference	416
6.244ngd_private Struct Reference	417
6.245nglmi_softc Struct Reference	419
6.246nglmistat Struct Reference	423
6.247ngm_atm_acr_change Struct Reference	424

6.248ngm_atm_config Struct Reference	425
6.249ngm_atm_cpcs_init Struct Reference	426
6.250ngm_atm_cpcs_term Struct Reference	429
6.251ngm_atm_if_change Struct Reference	430
6.252ngm_atm_stats Struct Reference	431
6.253ngm_atm_vcc_change Struct Reference	432
6.254ngm_bandwidth Struct Reference	433
6.255ngm_ccatm_addr_req Struct Reference	434
6.256ngm_ccatm_get_addresses Struct Reference	435
6.257ngm_ccatm_port Struct Reference	436
6.258ngm_ccatm_portlist Struct Reference	437
6.259ngm_connect Struct Reference	438
6.260ngm_mkpeer Struct Reference	439
6.261ngm_name Struct Reference	440
6.262ngm_queue_state Struct Reference	441
6.263ngm_rmhook Struct Reference	443
6.264ngm_uni_config_mask Struct Reference	444
6.265ngm_uni_debug Struct Reference	445
6.266ngm_uni_set_config Struct Reference	446
6.267ngm_vjc_config Struct Reference	447
6.268ngnf_flows Struct Reference	448
6.269ngpcb Struct Reference	449
6.270ngpppoe_init_data Struct Reference	451
6.271ngpppoe_sts Struct Reference	452
6.272ngpppoestat Struct Reference	453
6.273ngsock Struct Reference	454
6.274ngt_sc Struct Reference	456
6.275ngvcc Struct Reference	458
6.276ngxxxstat Struct Reference	460
6.277nodeinfo Struct Reference	461
6.278packet Union Reference	462
6.279PPPoE Struct Reference	463
6.280pppoe_full_hdr Struct Reference	465
6.281pppoe_hdr Struct Reference	466
6.282pppoe_tag Struct Reference	468
6.283priv Struct Reference	469

6.284priv_p Struct Reference	473
6.285private Struct Reference	475
6.286privdata Struct Reference	477
6.287protoent Struct Reference	480
6.288rfcomm_cmd_hdr Struct Reference	481
6.289rfcomm_frame_hdr Struct Reference	482
6.290rfcomm_mcc_hdr Struct Reference	483
6.291rfcomm_mcc_msc Struct Reference	484
6.292rfcomm_mcc_pn Struct Reference	485
6.293rfcomm_mcc_rls Struct Reference	487
6.294rfcomm_mcc_rpn Struct Reference	488
6.295sa_tag Struct Reference	490
6.296segment Struct Reference	491
6.297sess_con Struct Reference	492
6.298sess_neg Struct Reference	494
6.299sockaddr_hci Struct Reference	496
6.300sockaddr_l2cap Struct Reference	497
6.301sockaddr_ng Struct Reference	499
6.302sockaddr_rfcomm Struct Reference	500
6.303sscfu_arg Struct Reference	501
6.304sscop_arg Struct Reference	502
6.305sscop_marg Struct Reference	503
6.306sscop_merr Struct Reference	504
6.307stats Struct Reference	505
6.308typeinfo Struct Reference	507
6.309typelist Struct Reference	508
6.310ubt_softc Struct Reference	509
6.311ubtbcmfw_softc Struct Reference	516
6.312uni_arg Struct Reference	518
6.313uni_timer Struct Reference	519
6.314unimem_debug Struct Reference	520
6.315uniq Union Reference	521
6.316vatmpif_header Struct Reference	522
6.317vatmpif_stats Struct Reference	523
6.318vatmpif_unit Struct Reference	524
6.319vatmpif_vcc Struct Reference	526

6.320XXX Struct Reference	527
6.321XXX_hookinfo Struct Reference	529
7 FreeBSD kernel netgraph code File Documentation	531
7.1 notreviewed.dox File Reference	531
7.2 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif.c File Reference	532
7.3 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c File Reference	540
7.4 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h File Reference	546
7.5 /usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c File Reference	552
7.6 /usr/src/sys/netgraph/atm/ccatm/ng_ccatm_cust.h File Reference	565
7.7 /usr/src/sys/netgraph/atm/ng_atm.c File Reference	567
7.8 /usr/src/sys/netgraph/atm/ng_atm.h File Reference	583
7.9 /usr/src/sys/netgraph/atm/ng_atmpif.h File Reference	588
7.10 /usr/src/sys/netgraph/atm/ng_ccatm.h File Reference	593
7.11 /usr/src/sys/netgraph/atm/ng_sscfu.h File Reference	597
7.12 /usr/src/sys/netgraph/atm/ng_sscop.h File Reference	599
7.13 /usr/src/sys/netgraph/atm/ng_uni.h File Reference	602
7.14 /usr/src/sys/netgraph/atm/ngatmbase.c File Reference	605
7.15 /usr/src/sys/netgraph/atm/ngatmbase.h File Reference	609
7.16 /usr/src/sys/netgraph/atm/sscfu/ng_sscfu.c File Reference	611
7.17 /usr/src/sys/netgraph/atm/sscfu/ng_sscfu_cust.h File Reference	618
7.18 /usr/src/sys/netgraph/atm/sscop/ng_sscop.c File Reference	621
7.19 /usr/src/sys/netgraph/atm/sscop/ng_sscop_cust.h File Reference	630
7.20 /usr/src/sys/netgraph/atm/uni/ng_uni.c File Reference	639
7.21 /usr/src/sys/netgraph/atm/uni/ng_uni_cust.h File Reference	651
7.22 /usr/src/sys/netgraph/bluetooth/common/ng_bluetooth.c File Reference	657
7.23 /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_pccard.c File Reference	662
7.24 /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h File Reference	674
7.25 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4.c File Reference	678
7.26 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_prse.h File Reference	687
7.27 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_var.h File Reference	688
7.28 /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt.c File Reference	691
7.29 /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt_var.h File Reference	705
7.30 /usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ubtbcmfw.c File Reference	710
7.31 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.c File Reference	718
7.32 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.h File Reference	726
7.33 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.c File Reference	729

7.34	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.h File Reference	739
7.35	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_main.c File Reference	741
7.36	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.c File Reference	750
7.37	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.h File Reference	756
7.38	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_prse.h File Reference	761
7.39	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.c File Reference	764
7.40	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.h File Reference	771
7.41	/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h File Reference	776
7.42	/usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h File Reference	783
7.43	/usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h File Reference	791
7.44	/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h File Reference	795
7.45	/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_hci_raw.h File Reference	805
7.46	/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h File Reference	811
7.47	/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h File Reference	823
7.48	/usr/src/sys/netgraph/bluetooth/include/ng_h4.h File Reference	843
7.49	/usr/src/sys/netgraph/bluetooth/include/ng_hci.h File Reference	847
7.50	/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h File Reference	900
7.51	/usr/src/sys/netgraph/bluetooth/include/ng_ubt.h File Reference	919
7.52	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.c File Reference	922
7.53	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.h File Reference	927
7.54	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.c File Reference	934
7.55	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.h File Reference	944
7.56	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.c File Reference	946
7.57	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.h File Reference	953
7.58	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_main.c File Reference	959
7.59	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.c File Reference	968
7.60	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.h File Reference	977
7.61	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_prse.h File Reference	985
7.62	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.c File Reference	986
7.63	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h File Reference	997
7.64	/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h File Reference	1006
7.65	/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket.c File Reference	1010
7.66	/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_hci_raw.c File Reference	1016
7.67	/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap.c File Reference	1029
7.68	/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap_raw.c File Reference	1051
7.69	/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_rfcomm.c File Reference	1063

7.70	/usr/src/sys/netgraph/netflow/netflow.c File Reference	1090
7.71	/usr/src/sys/netgraph/netflow/netflow.h File Reference	1098
7.72	/usr/src/sys/netgraph/netflow/ng_netflow.c File Reference	1100
7.73	/usr/src/sys/netgraph/netflow/ng_netflow.h File Reference	1108
7.74	/usr/src/sys/netgraph/netgraph.h File Reference	1116
7.75	/usr/src/sys/netgraph/ng_async.c File Reference	1158
7.76	/usr/src/sys/netgraph/ng_async.h File Reference	1165
7.77	/usr/src/sys/netgraph/ng_atmllc.c File Reference	1168
7.78	/usr/src/sys/netgraph/ng_atmllc.h File Reference	1173
7.79	/usr/src/sys/netgraph/ng_base.c File Reference	1175
7.80	/usr/src/sys/netgraph/ng_bpf.c File Reference	1204
7.81	/usr/src/sys/netgraph/ng_bpf.h File Reference	1211
7.82	/usr/src/sys/netgraph/ng_bridge.c File Reference	1213
7.83	/usr/src/sys/netgraph/ng_bridge.h File Reference	1225
7.84	/usr/src/sys/netgraph/ng_cisco.c File Reference	1228
7.85	/usr/src/sys/netgraph/ng_cisco.h File Reference	1237
7.86	/usr/src/sys/netgraph/ng_deflate.c File Reference	1240
7.87	/usr/src/sys/netgraph/ng_deflate.h File Reference	1247
7.88	/usr/src/sys/netgraph/ng_device.c File Reference	1249
7.89	/usr/src/sys/netgraph/ng_device.h File Reference	1256
7.90	/usr/src/sys/netgraph/ng_echo.c File Reference	1257
7.91	/usr/src/sys/netgraph/ng_echo.h File Reference	1260
7.92	/usr/src/sys/netgraph/ng_eiface.c File Reference	1261
7.93	/usr/src/sys/netgraph/ng_eiface.h File Reference	1268
7.94	/usr/src/sys/netgraph/ng_etf.c File Reference	1270
7.95	/usr/src/sys/netgraph/ng_etf.h File Reference	1276
7.96	/usr/src/sys/netgraph/ng_ether.c File Reference	1278
7.97	/usr/src/sys/netgraph/ng_ether.h File Reference	1286
7.98	/usr/src/sys/netgraph/ng_fec.c File Reference	1288
7.99	/usr/src/sys/netgraph/ng_fec.h File Reference	1297
7.100	/usr/src/sys/netgraph/ng_frame_relay.c File Reference	1300
7.101	/usr/src/sys/netgraph/ng_frame_relay.h File Reference	1307
7.102	/usr/src/sys/netgraph/ng_gif.c File Reference	1308
7.103	/usr/src/sys/netgraph/ng_gif.h File Reference	1316
7.104	/usr/src/sys/netgraph/ng_gif_demux.c File Reference	1318
7.105	/usr/src/sys/netgraph/ng_gif_demux.h File Reference	1324

7.106/usr/src/sys/netgraph/ng_hole.c File Reference	1326
7.107/usr/src/sys/netgraph/ng_hole.h File Reference	1330
7.108/usr/src/sys/netgraph/ng_hub.c File Reference	1332
7.109/usr/src/sys/netgraph/ng_hub.h File Reference	1335
7.110/usr/src/sys/netgraph/ng_iface.c File Reference	1336
7.111/usr/src/sys/netgraph/ng_iface.h File Reference	1345
7.112/usr/src/sys/netgraph/ng_ip_input.c File Reference	1347
7.113/usr/src/sys/netgraph/ng_ip_input.h File Reference	1350
7.114/usr/src/sys/netgraph/ng_ipfw.c File Reference	1351
7.115/usr/src/sys/netgraph/ng_ipfw.h File Reference	1357
7.116/usr/src/sys/netgraph/ng_ksocket.c File Reference	1359
7.117/usr/src/sys/netgraph/ng_ksocket.h File Reference	1372
7.118/usr/src/sys/netgraph/ng_l2tp.c File Reference	1374
7.119/usr/src/sys/netgraph/ng_l2tp.h File Reference	1389
7.120/usr/src/sys/netgraph/ng_lmi.c File Reference	1393
7.121/usr/src/sys/netgraph/ng_lmi.h File Reference	1403
7.122/usr/src/sys/netgraph/ng_message.h File Reference	1406
7.123/usr/src/sys/netgraph/ng_mppc.c File Reference	1419
7.124/usr/src/sys/netgraph/ng_mppc.h File Reference	1427
7.125/usr/src/sys/netgraph/ng_nat.c File Reference	1430
7.126/usr/src/sys/netgraph/ng_nat.h File Reference	1435
7.127/usr/src/sys/netgraph/ng_one2many.c File Reference	1437
7.128/usr/src/sys/netgraph/ng_one2many.h File Reference	1443
7.129/usr/src/sys/netgraph/ng_parse.c File Reference	1446
7.130/usr/src/sys/netgraph/ng_parse.h File Reference	1475
7.131/usr/src/sys/netgraph/ng_ppp.c File Reference	1483
7.132/usr/src/sys/netgraph/ng_ppp.h File Reference	1511
7.133/usr/src/sys/netgraph/ng_pppoe.c File Reference	1517
7.134/usr/src/sys/netgraph/ng_pppoe.h File Reference	1530
7.135/usr/src/sys/netgraph/ng_pptpgre.c File Reference	1539
7.136/usr/src/sys/netgraph/ng_pptpgre.h File Reference	1551
7.137/usr/src/sys/netgraph/ng_pred1.c File Reference	1554
7.138/usr/src/sys/netgraph/ng_pred1.h File Reference	1562
7.139/usr/src/sys/netgraph/ng_rfc1490.c File Reference	1564
7.140/usr/src/sys/netgraph/ng_rfc1490.h File Reference	1571
7.141/usr/src/sys/netgraph/ng_sample.c File Reference	1573

7.142/usr/src/sys/netgraph/ng_sample.h File Reference	1578
7.143/usr/src/sys/netgraph/ng_socket.c File Reference	1580
7.144/usr/src/sys/netgraph/ng_socket.h File Reference	1593
7.145/usr/src/sys/netgraph/ng_socketvar.h File Reference	1595
7.146/usr/src/sys/netgraph/ng_source.c File Reference	1596
7.147/usr/src/sys/netgraph/ng_source.h File Reference	1605
7.148/usr/src/sys/netgraph/ng_split.c File Reference	1607
7.149/usr/src/sys/netgraph/ng_split.h File Reference	1611
7.150/usr/src/sys/netgraph/ng_sppp.c File Reference	1612
7.151/usr/src/sys/netgraph/ng_sppp.h File Reference	1619
7.152/usr/src/sys/netgraph/ng_tag.c File Reference	1621
7.153/usr/src/sys/netgraph/ng_tag.h File Reference	1628
7.154/usr/src/sys/netgraph/ng_tcpmss.c File Reference	1630
7.155/usr/src/sys/netgraph/ng_tcpmss.h File Reference	1636
7.156/usr/src/sys/netgraph/ng_tee.c File Reference	1638
7.157/usr/src/sys/netgraph/ng_tee.h File Reference	1643
7.158/usr/src/sys/netgraph/ng_tty.c File Reference	1645
7.159/usr/src/sys/netgraph/ng_tty.h File Reference	1654
7.160/usr/src/sys/netgraph/ng_UI.c File Reference	1656
7.161/usr/src/sys/netgraph/ng_UI.h File Reference	1660
7.162/usr/src/sys/netgraph/ng_vjc.c File Reference	1661
7.163/usr/src/sys/netgraph/ng_vjc.h File Reference	1668
7.164/usr/src/sys/netgraph/ng_vlan.c File Reference	1671
7.165/usr/src/sys/netgraph/ng_vlan.h File Reference	1678

Chapter 1

FreeBSD kernel netgraph code Main Page

IMPORTANT: This API documentation may contain both functions which are public and functions that are for internal use only. Since we have not reviewed every part of the documentation yet, *some internal functions are not marked as such*. Until we finish reviewing the API documentation and add appropriate comments to functions which are only for internal use, you should take this into account. In case you want to use a function of this kernel subsystem in another kernel subsystem you should search for precedence of use outside this subsystem. If the function is not used outside this subsystem you should ask on the mailinglists about it, else you risk breaking something.

Chapter 2

FreeBSD kernel netgraph code Directory Hierarchy

2.1 FreeBSD kernel netgraph code Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

usr	40
src	33
sys	36
netgraph	29
atm	17
atmpif	18
ccatm	21
sscfu	34
sscop	35
uni	39
bluetooth	19
common	22
drivers	23
bt3c	20
h4	24
ubt	37
ubtbcmfw	38
hci	25
include	26
l2cap	27
socket	32
netflow	28

Chapter 3

FreeBSD kernel netgraph code Data Structure Index

3.1 FreeBSD kernel netgraph code Data Structures

Here are the data structures with brief descriptions:

__attribute__	41
bt3c_softc	54
ccatm_op	58
chook	59
ccnode	61
cisco_header	63
cisco_packet	64
cisco_priv	66
ctxinfo	69
datatag	71
ETF	72
ETF_hookinfo	74
filter	75
flow_entry	76
flow_entry_data	77
flow_hash_entry	79
flow_manager	80
flow_rec	81
frmrel_softc	83
greheader	85
hookinfo	87
hooklist	88
hpriv_p	89
hva_stats_aal5	90
hva_stats_atm	92
hva_stats_ng	93
iffam	95
int16_temp	96
int32_temp	97
int64_temp	98
l2tp_seq	99

linkinfo	102
namelist	103
netflow	104
netflow_v1_header	106
netflow_v1_record	107
netflow_v5_export_dgram	110
netflow_v5_header	111
netflow_v5_record	113
ng_async_cfg	116
ng_async_private	117
ng_async_stat	120
ng_atmlc_priv	122
ng_atmpif_link_status	124
ng_bpf_hookinfo	125
ng_bpf_hookprog	127
ng_bpf_hookstat	128
ng_bridge_config	130
ng_bridge_hent	131
ng_bridge_host	132
ng_bridge_host_ary	133
ng_bridge_link	134
ng_bridge_link_stats	136
ng_bridge_private	139
ng_bt3c_firmware_block_ep	141
ng_bt3c_node_qlen_ep	142
ng_bt3c_node_stat_ep	143
ng_bt_itemq	144
ng_bt_mbufq	146
ng_btsocket_hci_raw_con_list	147
ng_btsocket_hci_raw_filter	148
ng_btsocket_hci_raw_node_bdaddr	149
ng_btsocket_hci_raw_node_buffer	150
ng_btsocket_hci_raw_node_debug	151
ng_btsocket_hci_raw_node_features	152
ng_btsocket_hci_raw_node_link_policy_mask	153
ng_btsocket_hci_raw_node_list_names	154
ng_btsocket_hci_raw_node_neighbor_cache	155
ng_btsocket_hci_raw_node_packet_mask	156
ng_btsocket_hci_raw_node_role_switch	157
ng_btsocket_hci_raw_node_stat	158
ng_btsocket_hci_raw_node_state	159
ng_btsocket_hci_raw_pcb	160
ng_btsocket_hci_raw_sec_filter	162
ng_btsocket_l2cap_pcb	163
ng_btsocket_l2cap_raw_auto_discon_timo	168
ng_btsocket_l2cap_raw_chan_list	169
ng_btsocket_l2cap_raw_con_list	170
ng_btsocket_l2cap_raw_get_info	171
ng_btsocket_l2cap_raw_node_debug	172
ng_btsocket_l2cap_raw_node_flags	173
ng_btsocket_l2cap_raw_pcb	174
ng_btsocket_l2cap_raw_ping	177
ng_btsocket_l2cap_rtentry	178
ng_btsocket_rfcomm_fc_info	180

ng_btsocket_rfcomm_pcb	182
ng_btsocket_rfcomm_session	186
ng_cisco_ipaddr	189
ng_cisco_stats	190
ng_cmdlist	191
ng_deflate_config	193
ng_deflate_private	194
ng_deflate_stats	196
ng_eiface_private	197
ng_etfilter	199
ng_etfstat	200
ng_fec_bundle	201
ng_fec_ifname	202
ng_fec_portlist	203
ng_fec_private	204
ng_gif_demux_private	206
ng_h4_info	208
ng_h4_node_stat_ep	211
ng_hci_lp_con_cfm_ep	212
ng_hci_lp_con_ind_ep	213
ng_hci_lp_con_req_ep	214
ng_hci_lp_con_rsp_ep	215
ng_hci_lp_discon_ind_ep	216
ng_hci_lp_discon_req_ep	217
ng_hci_lp_qos_cfm_ep	218
ng_hci_lp_qos_ind_ep	219
ng_hci_lp_qos_req_ep	220
ng_hci_neighbor	222
ng_hci_node_buffer_ep	224
ng_hci_node_con_ep	226
ng_hci_node_con_list_ep	228
ng_hci_node_get_neighbor_cache_ep	229
ng_hci_node_neighbor_cache_entry_ep	230
ng_hci_node_stat_ep	231
ng_hci_node_up_ep	233
ng_hci_sync_con_queue_ep	234
ng_hci_unit	235
ng_hci_unit_buff	239
ng_hci_unit_con	241
ng_hole_hookinfo	244
ng_hole_hookstat	245
ng_hook	246
ng_iface_private	249
ng_ipfw_hook_priv	251
ng_ipfw_tag	252
ng_item	253
ng_ksocket_accept	256
ng_ksocket_alias	257
ng_ksocket_private	258
ng_ksocket_sockopt	260
ng_l2cap	261
ng_l2cap_cfg_opt_val_t	264
ng_l2cap_chan	265
ng_l2cap_cmd	269

ng_l2cap_cmd_rej_data_t	271
ng_l2cap_con	272
ng_l2cap_info_rsp_data_t	276
ng_l2cap_l2ca_cfg_ind_ip	277
ng_l2cap_l2ca_cfg_ip	278
ng_l2cap_l2ca_cfg_op	279
ng_l2cap_l2ca_cfg_rsp_ip	280
ng_l2cap_l2ca_cfg_rsp_op	281
ng_l2cap_l2ca_con_ind_ip	282
ng_l2cap_l2ca_con_ip	283
ng_l2cap_l2ca_con_op	284
ng_l2cap_l2ca_con_rsp_ip	285
ng_l2cap_l2ca_con_rsp_op	286
ng_l2cap_l2ca_discon_ip	287
ng_l2cap_l2ca_discon_op	288
ng_l2cap_l2ca_enable_clt_ip	289
ng_l2cap_l2ca_get_info_ip	290
ng_l2cap_l2ca_get_info_op	291
ng_l2cap_l2ca_grp_add_member_ip	292
ng_l2cap_l2ca_grp_add_member_op	293
ng_l2cap_l2ca_grp_close_ip	294
ng_l2cap_l2ca_grp_create_ip	295
ng_l2cap_l2ca_grp_create_op	296
ng_l2cap_l2ca_grp_get_members_ip	297
ng_l2cap_l2ca_grp_get_members_op	298
ng_l2cap_l2ca_ping_ip	299
ng_l2cap_l2ca_ping_op	300
ng_l2cap_l2ca_qos_ind_ip	301
ng_l2cap_l2ca_write_op	302
ng_l2cap_node_chan_ep	303
ng_l2cap_node_chan_list_ep	305
ng_l2cap_node_con_ep	306
ng_l2cap_node_con_list_ep	307
ng_l2tp_config	308
ng_l2tp_hook_private	310
ng_l2tp_private	311
ng_l2tp_seq_config	313
ng_l2tp_sess_config	314
ng_l2tp_session_stats	315
ng_l2tp_stats	316
ng_mesg	319
ng_mesg::ng_msghdr	321
ng_mppc_config	324
ng_mppc_dir	325
ng_mppc_private	327
ng_nat_priv	328
ng_netflow_iface	330
ng_netflow_ifinfo	332
ng_netflow_info	333
ng_netflow_setdl	335
ng_netflow_setifindex	336
ng_netflow_settimeouts	337
ng_node	338
ng_one2many_config	341

ng_one2many_link	342
ng_one2many_link_stats	343
ng_one2many_private	344
ng_parse_array_info	346
ng_parse_fixedarray_info	347
ng_parse_fixedstring_info	348
ng_parse_struct_field	349
ng_parse_type	350
ng_ppp_bund_conf	352
ng_ppp_frag	355
ng_ppp_link	357
ng_ppp_link_conf	359
ng_ppp_link_stat	361
ng_ppp_mp_state	363
ng_ppp_node_conf	364
ng_ppp_private	365
ng_pptpgre_ackp	367
ng_pptpgre_conf	369
ng_pptpgre_private	371
ng_pptpgre_stats	373
ng_pred1_config	375
ng_pred1_private	376
ng_pred1_stats	378
ng_queue	379
ng_rfc1490_encap_t	381
ng_rfc1490_private	382
ng_source_stats	384
ng_split_private	386
ng_sppp_private	388
ng_sscfu_getdefparam	390
ng_sscop_setparam	391
ng_sscop_setparam_resp	392
ng_tag_hookin	393
ng_tag_hookinfo	395
ng_tag_hookout	397
ng_tag_prio	398
ng_tcpmss_config	399
ng_tcpmss_hookstat	400
ng_tee_hookstat	401
ng_tee_stats	402
ng_type	403
ng_ubt_node_qlen_ep	406
ng_ubt_node_stat_ep	407
ng_UI_private	408
ng_vatmpif_config	409
ng_vatmpif_hook	410
ng_vjc_private	412
ng_vlan_filter	414
ng_vlan_table	415
ngatm_msg	416
ngd_private	417
nglmi_softc	419
nglmistat	423
ngm_atm_acr_change	424

ngm_atm_config	425
ngm_atm_cpcs_init	426
ngm_atm_cpcs_term	429
ngm_atm_if_change	430
ngm_atm_stats	431
ngm_atm_vcc_change	432
ngm_bandwidth	433
ngm_ccatm_addr_req	434
ngm_ccatm_get_addresses	435
ngm_ccatm_port	436
ngm_ccatm_portlist	437
ngm_connect	438
ngm_mkpeer	439
ngm_name	440
ngm_queue_state	441
ngm_rmhook	443
ngm_uni_config_mask	444
ngm_uni_debug	445
ngm_uni_set_config	446
ngm_vjc_config	447
ngnf_flows	448
ngpcb	449
ngpppoe_init_data	451
ngpppoe_sts	452
ngpppoestat	453
ngsock	454
ngt_sc	456
ngvcc	458
ngxxxstat	460
nodeinfo	461
packet	462
PPPoE	463
pppoe_full_hdr	465
pppoe_hdr	466
pppoe_tag	468
priv	469
priv_p	473
private	475
privdata	477
protoent	480
rfcomm_cmd_hdr	481
rfcomm_frame_hdr	482
rfcomm_mcc_hdr	483
rfcomm_mcc_msc	484
rfcomm_mcc_pn	485
rfcomm_mcc_rls	487
rfcomm_mcc_rpn	488
sa_tag	490
segment	491
sess_con	492
sess_neg	494
sockaddr_hci	496
sockaddr_l2cap	497
sockaddr_ng	499

sockaddr_rfcomm	500
sscfu_arg	501
sscop_arg	502
sscop_marg	503
sscop_merr	504
stats	505
typeinfo	507
typelist	508
ubt_softc	509
ubtbcmfw_softc	516
uni_arg	518
uni_timer	519
unimem_debug	520
uniq	521
vatmpif_header	522
vatmpif_stats	523
vatmpif_unit	524
vatmpif_vcc	526
XXX	527
XXX_hookinfo	529

Chapter 4

FreeBSD kernel netgraph code File Index

4.1 FreeBSD kernel netgraph code File List

Here is a list of all files with brief descriptions:

/usr/src/sys/netgraph/netgraph.h	1116
/usr/src/sys/netgraph/ng_async.c	1158
/usr/src/sys/netgraph/ng_async.h	1165
/usr/src/sys/netgraph/ng_atmllc.c	1168
/usr/src/sys/netgraph/ng_atmllc.h	1173
/usr/src/sys/netgraph/ng_base.c	1175
/usr/src/sys/netgraph/ng_bpf.c	1204
/usr/src/sys/netgraph/ng_bpf.h	1211
/usr/src/sys/netgraph/ng_bridge.c	1213
/usr/src/sys/netgraph/ng_bridge.h	1225
/usr/src/sys/netgraph/ng_cisco.c	1228
/usr/src/sys/netgraph/ng_cisco.h	1237
/usr/src/sys/netgraph/ng_deflate.c	1240
/usr/src/sys/netgraph/ng_deflate.h	1247
/usr/src/sys/netgraph/ng_device.c	1249
/usr/src/sys/netgraph/ng_device.h	1256
/usr/src/sys/netgraph/ng_echo.c	1257
/usr/src/sys/netgraph/ng_echo.h	1260
/usr/src/sys/netgraph/ng_eiface.c	1261
/usr/src/sys/netgraph/ng_eiface.h	1268
/usr/src/sys/netgraph/ng_etf.c	1270
/usr/src/sys/netgraph/ng_etf.h	1276
/usr/src/sys/netgraph/ng_ether.c	1278
/usr/src/sys/netgraph/ng_ether.h	1286
/usr/src/sys/netgraph/ng_fec.c	1288
/usr/src/sys/netgraph/ng_fec.h	1297
/usr/src/sys/netgraph/ng_frame_relay.c	1300
/usr/src/sys/netgraph/ng_frame_relay.h	1307
/usr/src/sys/netgraph/ng_gif.c	1308
/usr/src/sys/netgraph/ng_gif.h	1316
/usr/src/sys/netgraph/ng_gif_demux.c	1318

/usr/src/sys/netgraph/ng_gif_demux.h	1324
/usr/src/sys/netgraph/ng_hole.c	1326
/usr/src/sys/netgraph/ng_hole.h	1330
/usr/src/sys/netgraph/ng_hub.c	1332
/usr/src/sys/netgraph/ng_hub.h	1335
/usr/src/sys/netgraph/ng_iface.c	1336
/usr/src/sys/netgraph/ng_iface.h	1345
/usr/src/sys/netgraph/ng_ip_input.c	1347
/usr/src/sys/netgraph/ng_ip_input.h	1350
/usr/src/sys/netgraph/ng_ipfw.c	1351
/usr/src/sys/netgraph/ng_ipfw.h	1357
/usr/src/sys/netgraph/ng_ksocket.c	1359
/usr/src/sys/netgraph/ng_ksocket.h	1372
/usr/src/sys/netgraph/ng_l2tp.c	1374
/usr/src/sys/netgraph/ng_l2tp.h	1389
/usr/src/sys/netgraph/ng_lmi.c	1393
/usr/src/sys/netgraph/ng_lmi.h	1403
/usr/src/sys/netgraph/ng_message.h	1406
/usr/src/sys/netgraph/ng_mppc.c	1419
/usr/src/sys/netgraph/ng_mppc.h	1427
/usr/src/sys/netgraph/ng_nat.c	1430
/usr/src/sys/netgraph/ng_nat.h	1435
/usr/src/sys/netgraph/ng_one2many.c	1437
/usr/src/sys/netgraph/ng_one2many.h	1443
/usr/src/sys/netgraph/ng_parse.c	1446
/usr/src/sys/netgraph/ng_parse.h	1475
/usr/src/sys/netgraph/ng_ppp.c	1483
/usr/src/sys/netgraph/ng_ppp.h	1511
/usr/src/sys/netgraph/ng_pppoe.c	1517
/usr/src/sys/netgraph/ng_pppoe.h	1530
/usr/src/sys/netgraph/ng_pptpgre.c	1539
/usr/src/sys/netgraph/ng_pptpgre.h	1551
/usr/src/sys/netgraph/ng_pred1.c	1554
/usr/src/sys/netgraph/ng_pred1.h	1562
/usr/src/sys/netgraph/ng_rfc1490.c	1564
/usr/src/sys/netgraph/ng_rfc1490.h	1571
/usr/src/sys/netgraph/ng_sample.c	1573
/usr/src/sys/netgraph/ng_sample.h	1578
/usr/src/sys/netgraph/ng_socket.c	1580
/usr/src/sys/netgraph/ng_socket.h	1593
/usr/src/sys/netgraph/ng_socketvar.h	1595
/usr/src/sys/netgraph/ng_source.c	1596
/usr/src/sys/netgraph/ng_source.h	1605
/usr/src/sys/netgraph/ng_split.c	1607
/usr/src/sys/netgraph/ng_split.h	1611
/usr/src/sys/netgraph/ng_sppp.c	1612
/usr/src/sys/netgraph/ng_sppp.h	1619
/usr/src/sys/netgraph/ng_tag.c	1621
/usr/src/sys/netgraph/ng_tag.h	1628
/usr/src/sys/netgraph/ng_tcpmss.c	1630
/usr/src/sys/netgraph/ng_tcpmss.h	1636
/usr/src/sys/netgraph/ng_tee.c	1638
/usr/src/sys/netgraph/ng_tee.h	1643
/usr/src/sys/netgraph/ng_tty.c	1645

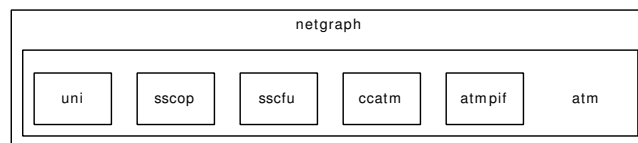
/usr/src/sys/netgraph/ng_tty.h	1654
/usr/src/sys/netgraph/ng_UI.c	1656
/usr/src/sys/netgraph/ng_UI.h	1660
/usr/src/sys/netgraph/ng_vjc.c	1661
/usr/src/sys/netgraph/ng_vjc.h	1668
/usr/src/sys/netgraph/ng_vlan.c	1671
/usr/src/sys/netgraph/ng_vlan.h	1678
/usr/src/sys/netgraph/atm/ng_atm.c	567
/usr/src/sys/netgraph/atm/ng_atm.h	583
/usr/src/sys/netgraph/atm/ng_atmpif.h	588
/usr/src/sys/netgraph/atm/ng_ccatm.h	593
/usr/src/sys/netgraph/atm/ng_sscfu.h	597
/usr/src/sys/netgraph/atm/ng_sscop.h	599
/usr/src/sys/netgraph/atm/ng_uni.h	602
/usr/src/sys/netgraph/atm/ngatmbase.c	605
/usr/src/sys/netgraph/atm/ngatmbase.h	609
/usr/src/sys/netgraph/atm/atmpif/ng_atmpif.c	532
/usr/src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c	540
/usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h	546
/usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c	552
/usr/src/sys/netgraph/atm/ccatm/ng_ccatm_cust.h	565
/usr/src/sys/netgraph/atm/sscfu/ng_sscfu.c	611
/usr/src/sys/netgraph/atm/sscfu/ng_sscfu_cust.h	618
/usr/src/sys/netgraph/atm/sscop/ng_sscop.c	621
/usr/src/sys/netgraph/atm/sscop/ng_sscop_cust.h	630
/usr/src/sys/netgraph/atm/uni/ng_uni.c	639
/usr/src/sys/netgraph/atm/uni/ng_uni_cust.h	651
/usr/src/sys/netgraph/bluetooth/common/ng_bluetooth.c	657
/usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_pccard.c	662
/usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h	674
/usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4.c	678
/usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_prse.h	687
/usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_var.h	688
/usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt.c	691
/usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt_var.h	705
/usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ubtbcmfw.c	710
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.c	718
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.h	726
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.c	729
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.h	739
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_main.c	741
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.c	750
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.h	756
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_prse.h	761
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.c	764
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.h	771
/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h	776
/usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h	783
/usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h	791
/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h	795
/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_hci_raw.h	805
/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h	811
/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h	823
/usr/src/sys/netgraph/bluetooth/include/ng_h4.h	843

/usr/src/sys/netgraph/bluetooth/include/ng_hci.h	847
/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h	900
/usr/src/sys/netgraph/bluetooth/include/ng_ubt.h	919
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.c	922
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.h	927
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.c	934
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.h	944
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.c	946
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.h	953
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_main.c	959
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.c	968
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.h	977
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_prse.h	985
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.c	986
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h	997
/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h	1006
/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket.c	1010
/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_hci_raw.c	1016
/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap.c	1029
/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap_raw.c	1051
/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_rfcomm.c	1063
/usr/src/sys/netgraph/netflow/netflow.c	1090
/usr/src/sys/netgraph/netflow/netflow.h	1098
/usr/src/sys/netgraph/netflow/ng_netflow.c	1100
/usr/src/sys/netgraph/netflow/ng_netflow.h	1108

Chapter 5

FreeBSD kernel netgraph code Directory Documentation

5.1 /usr/src/sys/netgraph/atm/ Directory Reference



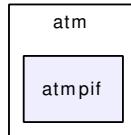
Directories

- directory [atmpif](#)
- directory [ccatm](#)
- directory [sscfu](#)
- directory [sscop](#)
- directory [uni](#)

Files

- file [ng_atm.c](#)
- file [ng_atm.h](#)
- file [ng_atmpif.h](#)
- file [ng_ccatm.h](#)
- file [ng_sscfu.h](#)
- file [ng_sscop.h](#)
- file [ng_uni.h](#)
- file [ngatmbase.c](#)
- file [ngatmbase.h](#)

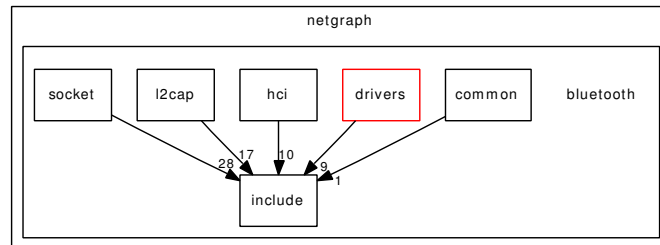
5.2 /usr/src/sys/netgraph/atm/atmpif/ Directory Reference



Files

- file [ng_atmpif.c](#)
- file [ng_atmpif_harp.c](#)
- file [ng_atmpif_var.h](#)

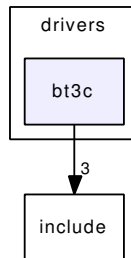
5.3 /usr/src/sys/netgraph/bluetooth/ Directory Reference



Directories

- directory [common](#)
- directory [drivers](#)
- directory [hci](#)
- directory [include](#)
- directory [l2cap](#)
- directory [socket](#)

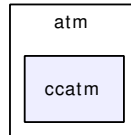
5.4 /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ Directory Reference



Files

- file [ng_bt3c_pccard.c](#)
- file [ng_bt3c_var.h](#)

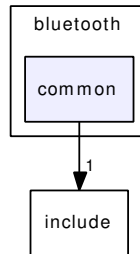
5.5 /usr/src/sys/netgraph/atm/ccatm/ Directory Reference



Files

- file [ng_ccatm.c](#)
- file [ng_ccatm_cust.h](#)

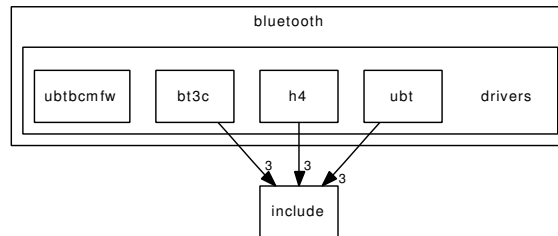
5.6 /usr/src/sys/netgraph/bluetooth/common/ Directory Reference



Files

- file [ng_bluetooth.c](#)

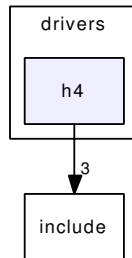
5.7 /usr/src/sys/netgraph/bluetooth/drivers/ Directory Reference



Directories

- directory [bt3c](#)
- directory [h4](#)
- directory [ubt](#)
- directory [ubtbcmfw](#)

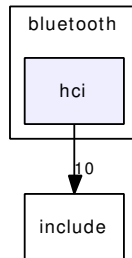
5.8 /usr/src/sys/netgraph/bluetooth/drivers/h4/ Directory Reference



Files

- file [ng_h4.c](#)
- file [ng_h4_prse.h](#)
- file [ng_h4_var.h](#)

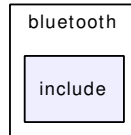
5.9 /usr/src/sys/netgraph/bluetooth/hci/ Directory Reference



Files

- file [ng_hci_cmds.c](#)
- file [ng_hci_cmds.h](#)
- file [ng_hci_evnt.c](#)
- file [ng_hci_evnt.h](#)
- file [ng_hci_main.c](#)
- file [ng_hci_misc.c](#)
- file [ng_hci_misc.h](#)
- file [ng_hci_prse.h](#)
- file [ng_hci_ulpi.c](#)
- file [ng_hci_ulpi.h](#)
- file [ng_hci_var.h](#)

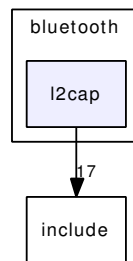
5.10 /usr/src/sys/netgraph/bluetooth/include/ Directory Reference



Files

- file [ng_bluetooth.h](#)
- file [ng_bt3c.h](#)
- file [ng_btsocket.h](#)
- file [ng_btsocket_hci_raw.h](#)
- file [ng_btsocket_l2cap.h](#)
- file [ng_btsocket_rfcomm.h](#)
- file [ng_h4.h](#)
- file [ng_hci.h](#)
- file [ng_l2cap.h](#)
- file [ng_ubt.h](#)

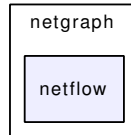
5.11 /usr/src/sys/netgraph/bluetooth/l2cap/ Directory Reference



Files

- file [ng_l2cap_cmds.c](#)
- file [ng_l2cap_cmds.h](#)
- file [ng_l2cap_evnt.c](#)
- file [ng_l2cap_evnt.h](#)
- file [ng_l2cap_llpi.c](#)
- file [ng_l2cap_llpi.h](#)
- file [ng_l2cap_main.c](#)
- file [ng_l2cap_misc.c](#)
- file [ng_l2cap_misc.h](#)
- file [ng_l2cap_prse.h](#)
- file [ng_l2cap_ulpi.c](#)
- file [ng_l2cap_ulpi.h](#)
- file [ng_l2cap_var.h](#)

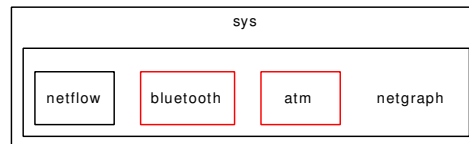
5.12 /usr/src/sys/netgraph/netflow/ Directory Reference



Files

- file [netflow.c](#)
- file [netflow.h](#)
- file [ng_netflow.c](#)
- file [ng_netflow.h](#)

5.13 /usr/src/sys/netgraph/ Directory Reference



Directories

- directory [atm](#)
- directory [bluetooth](#)
- directory [netflow](#)

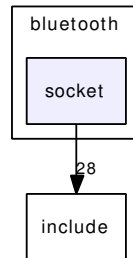
Files

- file [netgraph.h](#)
- file [ng_async.c](#)
- file [ng_async.h](#)
- file [ng_atmllc.c](#)
- file [ng_atmllc.h](#)
- file [ng_base.c](#)
- file [ng_bpf.c](#)
- file [ng_bpf.h](#)
- file [ng_bridge.c](#)
- file [ng_bridge.h](#)
- file [ng_cisco.c](#)
- file [ng_cisco.h](#)
- file [ng_deflate.c](#)
- file [ng_deflate.h](#)
- file [ng_device.c](#)
- file [ng_device.h](#)
- file [ng_echo.c](#)
- file [ng_echo.h](#)
- file [ng_eiface.c](#)
- file [ng_eiface.h](#)
- file [ng_etf.c](#)
- file [ng_etf.h](#)
- file [ng_ether.c](#)
- file [ng_ether.h](#)
- file [ng_fec.c](#)
- file [ng_fec.h](#)
- file [ng_frame_relay.c](#)
- file [ng_frame_relay.h](#)
- file [ng_gif.c](#)
- file [ng_gif.h](#)
- file [ng_gif_demux.c](#)

- file [ng_gif_demux.h](#)
- file [ng_hole.c](#)
- file [ng_hole.h](#)
- file [ng_hub.c](#)
- file [ng_hub.h](#)
- file [ng_iface.c](#)
- file [ng_iface.h](#)
- file [ng_ip_input.c](#)
- file [ng_ip_input.h](#)
- file [ng_ipfw.c](#)
- file [ng_ipfw.h](#)
- file [ng_ksocket.c](#)
- file [ng_ksocket.h](#)
- file [ng_l2tp.c](#)
- file [ng_l2tp.h](#)
- file [ng_lmi.c](#)
- file [ng_lmi.h](#)
- file [ng_message.h](#)
- file [ng_mppc.c](#)
- file [ng_mppc.h](#)
- file [ng_nat.c](#)
- file [ng_nat.h](#)
- file [ng_one2many.c](#)
- file [ng_one2many.h](#)
- file [ng_parse.c](#)
- file [ng_parse.h](#)
- file [ng_ppp.c](#)
- file [ng_ppp.h](#)
- file [ng_pppoe.c](#)
- file [ng_pppoe.h](#)
- file [ng_pptpgre.c](#)
- file [ng_pptpgre.h](#)
- file [ng_pred1.c](#)
- file [ng_pred1.h](#)
- file [ng_rfc1490.c](#)
- file [ng_rfc1490.h](#)
- file [ng_sample.c](#)
- file [ng_sample.h](#)
- file [ng_socket.c](#)
- file [ng_socket.h](#)
- file [ng_socketvar.h](#)
- file [ng_source.c](#)
- file [ng_source.h](#)
- file [ng_split.c](#)
- file [ng_split.h](#)
- file [ng_sppp.c](#)
- file [ng_sppp.h](#)
- file [ng_tag.c](#)
- file [ng_tag.h](#)
- file [ng_tcpmss.c](#)

- file [ng_tcpmss.h](#)
- file [ng_tee.c](#)
- file [ng_tee.h](#)
- file [ng_tty.c](#)
- file [ng_tty.h](#)
- file [ng_UI.c](#)
- file [ng_UI.h](#)
- file [ng_vjc.c](#)
- file [ng_vjc.h](#)
- file [ng_vlan.c](#)
- file [ng_vlan.h](#)

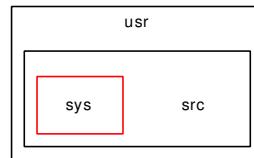
5.14 /usr/src/sys/netgraph/bluetooth/socket/ Directory Reference



Files

- file [ng_btsocket.c](#)
- file [ng_btsocket_hci_raw.c](#)
- file [ng_btsocket_l2cap.c](#)
- file [ng_btsocket_l2cap_raw.c](#)
- file [ng_btsocket_rfcomm.c](#)

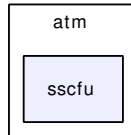
5.15 /usr/src/ Directory Reference



Directories

- directory [sys](#)

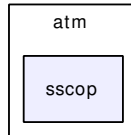
5.16 /usr/src/sys/netgraph/atm/sscfu/ Directory Reference



Files

- file [ng_sscfu.c](#)
- file [ng_sscfu_cust.h](#)

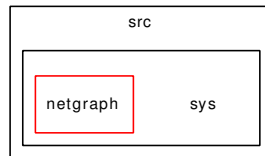
5.17 /usr/src/sys/netgraph/atm/sscop/ Directory Reference



Files

- file [ng_sscop.c](#)
- file [ng_sscop_cust.h](#)

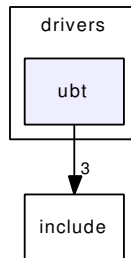
5.18 /usr/src/sys/ Directory Reference



Directories

- directory [netgraph](#)

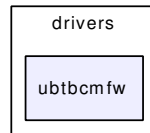
5.19 /usr/src/sys/netgraph/bluetooth/drivers/ubt/ Directory Reference



Files

- file [ng_ubt.c](#)
- file [ng_ubt_var.h](#)

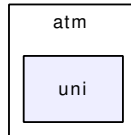
5.20 /usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ Directory Reference



Files

- file [ubtbcmfw.c](#)

5.21 /usr/src/sys/netgraph/atm/uni/ Directory Reference



Files

- file [ng_uni.c](#)
- file [ng_uni_cust.h](#)

5.22 /usr/ Directory Reference



Directories

- [directory src](#)

Chapter 6

FreeBSD kernel netgraph code Data Structure Documentation

6.1 `__attribute__` Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int8_t type`
- `u_int16_t opcode`
- `u_int8_t length`
- `u_int16_t con_handle`
- `u_int16_t length`
- `u_int8_t event`
- `u_int8_t b` [NG_HCI_BDADDR_SIZE]
- `u_int8_t status`
- `u_int8_t lap` [NG_HCI_LAP_SIZE]
- `u_int8_t inquiry_length`
- `u_int8_t num_responses`
- `u_int16_t max_period_length`
- `u_int16_t min_period_length`
- `bdaddr_t bdaddr`
- `u_int16_t pkt_type`
- `u_int8_t page_scan_rep_mode`
- `u_int8_t page_scan_mode`
- `u_int16_t clock_offset`
- `u_int8_t accept_role_switch`
- `u_int8_t reason`
- `u_int8_t role`
- `u_int8_t key` [NG_HCI_KEY_SIZE]
- `u_int8_t pin_size`
- `u_int8_t pin` [NG_HCI_PIN_SIZE]
- `u_int8_t encryption_enable`
- `u_int8_t key_flag`

- u_int16_t max_interval
- u_int16_t min_interval
- u_int16_t attempt
- u_int16_t timeout
- u_int8_t flags
- u_int8_t service_type
- u_int32_t token_rate
- u_int32_t peak_bandwidth
- u_int32_t latency
- u_int32_t delay_variation
- u_int16_t settings
- u_int8_t event_mask [NG_HCI_EVENT_MASK_SIZE]
- u_int8_t filter_type
- u_int8_t filter_condition_type
- u_int8_t condition [0]
- u_int8_t pin_type
- u_int8_t read_all
- u_int16_t max_num_keys
- u_int16_t num_keys_read
- u_int8_t num_keys_write
- u_int8_t num_keys_written
- u_int8_t delete_all
- u_int16_t num_keys_deleted
- char name [NG_HCI_UNIT_NAME_SIZE]
- u_int8_t scan_enable
- u_int16_t page_scan_interval
- u_int16_t page_scan_window
- u_int16_t inquiry_scan_interval
- u_int16_t inquiry_scan_window
- u_int8_t auth_enable
- u_int8_t encryption_mode
- u_int8_t uclass [NG_HCI_CLASS_SIZE]
- u_int8_t counter
- u_int8_t hold_mode_activity
- char level
- u_int8_t flow_control
- u_int8_t h2hc_flow
- u_int16_t max_acl_size
- u_int8_t max_sco_size
- u_int16_t num_acl_pkt
- u_int16_t num_sco_pkt
- u_int8_t num_con_handles
- u_int8_t num_iac
- u_int8_t page_scan_period_mode
- u_int8_t hci_version
- u_int16_t hci_revision
- u_int8_t lmp_version
- u_int16_t manufacturer
- u_int16_t lmp_subversion
- u_int8_t features [NG_HCI_FEATURES_SIZE]

- `u_int8_t` `country_code`
- `u_int16_t` `counter`
- `u_int8_t` `quality`
- `char` `rss`
- `u_int8_t` `lbmode`
- `u_int8_t` `link_type`
- `u_int8_t` `num_cmd_pkts`
- `u_int8_t` `hardware_code`
- `u_int8_t` `unit_mode`
- `u_int16_t` `interval`
- `u_int8_t` `num_keys`
- `u_int8_t` `key_type`
- `u_int8_t` `command` [0]
- `u_int8_t` `lmp_max_slots`
- `u_int32_t` `token_bucket_size`
- `u_int16_t` `dcid`
- `u_int16_t` `psm`
- `u_int8_t` `code`
- `u_int8_t` `ident`
- `u_int16_t` `reason`
- `u_int16_t` `scid`
- `u_int16_t` `result`
- `u_int16_t` `status`
- `u_int16_t` `flags`
- `u_int16_t` `type`
- `u_int32_t` `token`
- `u_int16_t` `lcid`

6.1.1 Detailed Description

Definition at line 340 of file `ng_hci.h`.

6.1.2 Field Documentation

6.1.2.1 `u_int8_t` `__attribute__`::`accept_role_switch`

Definition at line 651 of file `ng_hci.h`.

6.1.2.2 `u_int16_t` `__attribute__`::`attempt`

Definition at line 806 of file `ng_hci.h`.

6.1.2.3 `u_int8_t` `__attribute__`::`auth_enable`

Definition at line 1067 of file `ng_hci.h`.

6.1.2.4 `u_int8_t` `__attribute__`::`b`[`NG_HCI_BDADDR_SIZE`]

Definition at line 375 of file `ng_hci.h`.

6.1.2.5 `bdaddr_t __attribute__::bdaddr`

Definition at line 646 of file `ng_hci.h`.

6.1.2.6 `u_int16_t __attribute__::clock_offset`

Definition at line 650 of file `ng_hci.h`.

6.1.2.7 `u_int8_t __attribute__::code`

Definition at line 180 of file `ng_l2cap.h`.

6.1.2.8 `u_int8_t __attribute__::command[0]`

Definition at line 1613 of file `ng_hci.h`.

6.1.2.9 `u_int16_t __attribute__::con_handle`

Definition at line 351 of file `ng_hci.h`.

6.1.2.10 `u_int8_t __attribute__::condition[0]`

Definition at line 906 of file `ng_hci.h`.

6.1.2.11 `u_int16_t __attribute__::counter`

Definition at line 1358 of file `ng_hci.h`.

6.1.2.12 `u_int8_t __attribute__::counter`

Definition at line 1145 of file `ng_hci.h`.

6.1.2.13 `u_int8_t __attribute__::country_code`

Definition at line 1333 of file `ng_hci.h`.

6.1.2.14 `u_int16_t __attribute__::dcid`

Definition at line 167 of file `ng_l2cap.h`.

6.1.2.15 `u_int32_t __attribute__::delay_variation`

Definition at line 839 of file `ng_hci.h`.

6.1.2.16 `u_int8_t __attribute__::delete_all`

Definition at line 967 of file `ng_hci.h`.

6.1.2.17 `u_int8_t __attribute__::encryption_enable`

Definition at line 742 of file `ng_hci.h`.

6.1.2.18 `u_int8_t __attribute__::encryption_mode`

Definition at line 1081 of file `ng_hci.h`.

6.1.2.19 `u_int8_t __attribute__::event`

Definition at line 369 of file `ng_hci.h`.

6.1.2.20 `u_int8_t __attribute__::event_mask[NG_HCI_EVENT_MASK_SIZE]`

Definition at line 893 of file `ng_hci.h`.

6.1.2.21 `u_int8_t __attribute__::features`

Definition at line 1318 of file `ng_hci.h`.

6.1.2.22 `u_int8_t __attribute__::filter_condition_type`

Definition at line 905 of file `ng_hci.h`.

6.1.2.23 `u_int8_t __attribute__::filter_type`

Definition at line 904 of file `ng_hci.h`.

6.1.2.24 `u_int16_t __attribute__::flags`

Definition at line 226 of file `ng_l2cap.h`.

6.1.2.25 `u_int16_t __attribute__::flags`

Definition at line 834 of file `ng_hci.h`.

6.1.2.26 `u_int8_t __attribute__::flow_control`

Definition at line 1185 of file `ng_hci.h`.

6.1.2.27 `u_int8_t __attribute__((h2hc_flow))`

Definition at line 1197 of file ng_hci.h.

6.1.2.28 `u_int8_t __attribute__((hardware_code))`

Definition at line 1555 of file ng_hci.h.

6.1.2.29 `u_int16_t __attribute__((hci_revision))`

Definition at line 1309 of file ng_hci.h.

6.1.2.30 `u_int8_t __attribute__((hci_version))`

Definition at line 1308 of file ng_hci.h.

6.1.2.31 `u_int8_t __attribute__((hold_mode_activity))`

Definition at line 1159 of file ng_hci.h.

6.1.2.32 `u_int8_t __attribute__((ident))`

Definition at line 181 of file ng_l2cap.h.

6.1.2.33 `u_int8_t __attribute__((inquiry_length))`

Definition at line 620 of file ng_hci.h.

6.1.2.34 `u_int16_t __attribute__((inquiry_scan_interval))`

Definition at line 1051 of file ng_hci.h.

6.1.2.35 `u_int16_t __attribute__((inquiry_scan_window))`

Definition at line 1052 of file ng_hci.h.

6.1.2.36 `u_int16_t __attribute__((interval))`

Definition at line 1583 of file ng_hci.h.

6.1.2.37 `u_int8_t __attribute__((key))`

Definition at line 686 of file ng_hci.h.

6.1.2.38 `u_int8_t __attribute__::key_flag`

Definition at line 754 of file `ng_hci.h`.

6.1.2.39 `u_int8_t __attribute__::key_type`

Definition at line 1608 of file `ng_hci.h`.

6.1.2.40 `u_int8_t __attribute__::lap`

Definition at line 619 of file `ng_hci.h`.

6.1.2.41 `u_int32_t __attribute__::latency`

Definition at line 838 of file `ng_hci.h`.

6.1.2.42 `u_int8_t __attribute__::lbmode`

Definition at line 1405 of file `ng_hci.h`.

6.1.2.43 `u_int16_t __attribute__::lcid`

Definition at line 334 of file `ng_l2cap.h`.

6.1.2.44 `u_int16_t __attribute__::length`

Definition at line 352 of file `ng_hci.h`.

6.1.2.45 `u_int16_t __attribute__::length`

Definition at line 343 of file `ng_hci.h`.

6.1.2.46 `char __attribute__::level`

Definition at line 1178 of file `ng_hci.h`.

6.1.2.47 `u_int8_t __attribute__::link_type`

Definition at line 1460 of file `ng_hci.h`.

6.1.2.48 `u_int8_t __attribute__::lmp_max_slots`

Definition at line 1624 of file `ng_hci.h`.

6.1.2.49 `u_int16_t __attribute__((lmp_subversion))`

Definition at line 1312 of file ng_hci.h.

6.1.2.50 `u_int8_t __attribute__((lmp_version))`

Definition at line 1310 of file ng_hci.h.

6.1.2.51 `u_int16_t __attribute__((manufacturer))`

Definition at line 1311 of file ng_hci.h.

6.1.2.52 `u_int16_t __attribute__((max_acl_size))`

Definition at line 1204 of file ng_hci.h.

6.1.2.53 `u_int16_t __attribute__((max_interval))`

Definition at line 796 of file ng_hci.h.

6.1.2.54 `u_int16_t __attribute__((max_num_keys))`

Definition at line 947 of file ng_hci.h.

6.1.2.55 `u_int16_t __attribute__((max_period_length))`

Definition at line 631 of file ng_hci.h.

6.1.2.56 `u_int8_t __attribute__((max_sco_size))`

Definition at line 1205 of file ng_hci.h.

6.1.2.57 `u_int16_t __attribute__((min_interval))`

Definition at line 797 of file ng_hci.h.

6.1.2.58 `u_int16_t __attribute__((min_period_length))`

Definition at line 632 of file ng_hci.h.

6.1.2.59 `char __attribute__((name))`

Definition at line 977 of file ng_hci.h.

6.1.2.60 `u_int16_t __attribute__::num_acl_pkt`

Definition at line 1206 of file `ng_hci.h`.

6.1.2.61 `u_int8_t __attribute__::num_cmd_pkts`

Definition at line 1541 of file `ng_hci.h`.

6.1.2.62 `u_int8_t __attribute__::num_con_handles`

Definition at line 1214 of file `ng_hci.h`.

6.1.2.63 `u_int8_t __attribute__::num_iac`

Definition at line 1247 of file `ng_hci.h`.

6.1.2.64 `u_int8_t __attribute__::num_keys`

Definition at line 1588 of file `ng_hci.h`.

6.1.2.65 `u_int16_t __attribute__::num_keys_deleted`

Definition at line 972 of file `ng_hci.h`.

6.1.2.66 `u_int16_t __attribute__::num_keys_read`

Definition at line 948 of file `ng_hci.h`.

6.1.2.67 `u_int8_t __attribute__::num_keys_write`

Definition at line 953 of file `ng_hci.h`.

6.1.2.68 `u_int8_t __attribute__::num_keys_written`

Definition at line 961 of file `ng_hci.h`.

6.1.2.69 `u_int8_t __attribute__::num_responses`

Definition at line 621 of file `ng_hci.h`.

6.1.2.70 `u_int16_t __attribute__::num_sco_pkt`

Definition at line 1207 of file `ng_hci.h`.

6.1.2.71 `u_int16_t __attribute__((opcode))`

Definition at line 342 of file ng_hci.h.

6.1.2.72 `u_int16_t __attribute__((page_scan_interval))`

Definition at line 1035 of file ng_hci.h.

6.1.2.73 `u_int8_t __attribute__((page_scan_mode))`

Definition at line 649 of file ng_hci.h.

6.1.2.74 `u_int8_t __attribute__((page_scan_period_mode))`

Definition at line 1272 of file ng_hci.h.

6.1.2.75 `u_int8_t __attribute__((page_scan_rep_mode))`

Definition at line 648 of file ng_hci.h.

6.1.2.76 `u_int16_t __attribute__((page_scan_window))`

Definition at line 1036 of file ng_hci.h.

6.1.2.77 `u_int32_t __attribute__((peak_bandwidth))`

Definition at line 837 of file ng_hci.h.

6.1.2.78 `u_int8_t __attribute__((pin[NG_HCI_PIN_SIZE]))`

Definition at line 708 of file ng_hci.h.

6.1.2.79 `u_int8_t __attribute__((pin_size))`

Definition at line 707 of file ng_hci.h.

6.1.2.80 `u_int8_t __attribute__((pin_type))`

Definition at line 925 of file ng_hci.h.

6.1.2.81 `u_int16_t __attribute__((pkt_type))`

Definition at line 647 of file ng_hci.h.

6.1.2.82 `u_int16_t __attribute__::psm`

Definition at line 172 of file `ng_l2cap.h`.

6.1.2.83 `u_int8_t __attribute__::quality`

Definition at line 1379 of file `ng_hci.h`.

6.1.2.84 `u_int8_t __attribute__::read_all`

Definition at line 942 of file `ng_hci.h`.

6.1.2.85 `u_int16_t __attribute__::reason`

Definition at line 188 of file `ng_l2cap.h`.

6.1.2.86 `u_int8_t __attribute__::reason`

Definition at line 658 of file `ng_hci.h`.

6.1.2.87 `u_int16_t __attribute__::result`

Definition at line 218 of file `ng_l2cap.h`.

6.1.2.88 `u_int8_t __attribute__::role`

Definition at line 672 of file `ng_hci.h`.

6.1.2.89 `char __attribute__::rssi`

Definition at line 1390 of file `ng_hci.h`.

6.1.2.90 `u_int8_t __attribute__::scan_enable`

Definition at line 1021 of file `ng_hci.h`.

6.1.2.91 `u_int16_t __attribute__::scid`

Definition at line 210 of file `ng_l2cap.h`.

6.1.2.92 `u_int8_t __attribute__::service_type`

Definition at line 835 of file `ng_hci.h`.

6.1.2.93 `u_int16_t __attribute__::settings`

Definition at line 869 of file `ng_hci.h`.

6.1.2.94 `u_int16_t __attribute__::status`

Definition at line 219 of file `ng_l2cap.h`.

6.1.2.95 `u_int8_t __attribute__::status`

Definition at line 384 of file `ng_hci.h`.

6.1.2.96 `u_int16_t __attribute__::timeout`

Definition at line 807 of file `ng_hci.h`.

6.1.2.97 `u_int32_t __attribute__::token`

Definition at line 332 of file `ng_l2cap.h`.

6.1.2.98 `u_int32_t __attribute__::token_bucket_size`

Definition at line 151 of file `ng_l2cap.h`.

6.1.2.99 `u_int32_t __attribute__::token_rate`

Definition at line 836 of file `ng_hci.h`.

6.1.2.100 `u_int16_t __attribute__::type`

Definition at line 279 of file `ng_l2cap.h`.

6.1.2.101 `u_int16_t __attribute__::type`

Definition at line 341 of file `ng_hci.h`.

6.1.2.102 `u_int8_t __attribute__::uclass`

Definition at line 1095 of file `ng_hci.h`.

6.1.2.103 `u_int8_t __attribute__::unit_mode`

Definition at line 1582 of file `ng_hci.h`.

The documentation for this struct was generated from the following files:

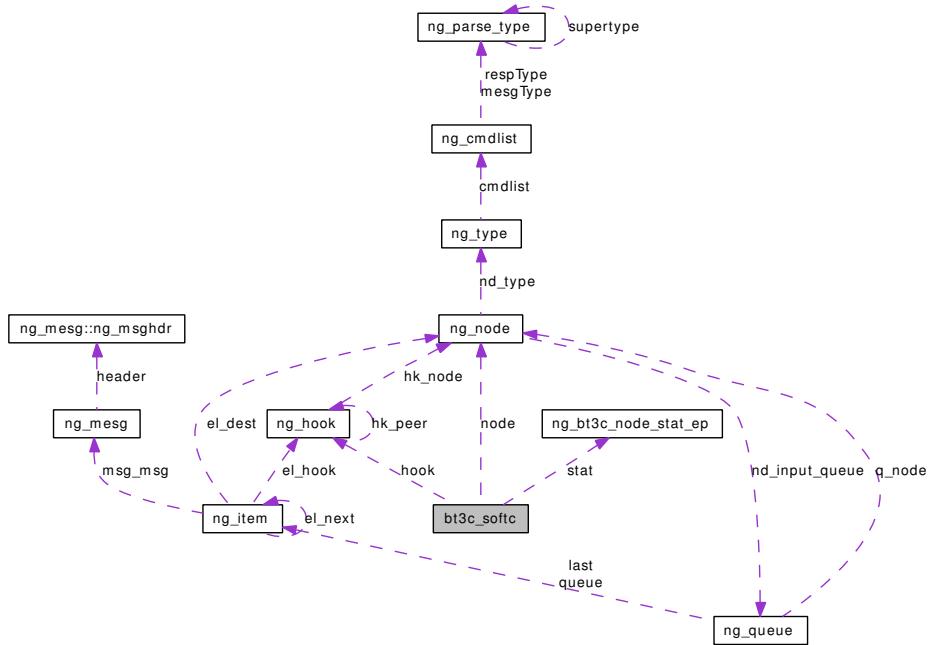
- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.2 bt3c_softc Struct Reference

```
#include <ng_bt3c_var.h>
```

Collaboration diagram for bt3c_softc:



Data Fields

- `device_t dev`
- `int iobase_rid`
- `resource * iobase`
- `bus_space_tag_t iot`
- `bus_space_handle_t ioh`
- `int irq_rid`
- `resource * irq`
- `void * irq_cookie`
- `node_p node`
- `hook_p hook`
- `ng_bt3c_node_debug_ep debug`
- `u_int16_t flags`
- `ng_bt3c_node_state_ep state`
- `ng_bt3c_node_stat_ep stat`
- `u_int32_t status`
- `void * ith`
- `mbuf * m`
- `u_int32_t want`
- `ifqueue inq`
- `ifqueue outq`

6.2.1 Detailed Description

Definition at line 61 of file ng_bt3c_var.h.

6.2.2 Field Documentation

6.2.2.1 [ng_bt3c_node_debug_ep bt3c_softc::debug](#)

Definition at line 76 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and ng_bt3c_rcvmsg().

6.2.2.2 [device_t bt3c_softc::dev](#)

Definition at line 63 of file ng_bt3c_var.h.

Referenced by bt3c_download_firmware(), bt3c_intr(), bt3c_pccard_attach(), bt3c_receive(), bt3c_send(), bt3c_swi_intr(), ng_bt3c_rcvdata(), and ng_bt3c_shutdown().

6.2.2.3 [u_int16_t bt3c_softc::flags](#)

Definition at line 77 of file ng_bt3c_var.h.

Referenced by bt3c_send(), bt3c_swi_intr(), and ng_bt3c_rcvmsg().

6.2.2.4 [hook_p bt3c_softc::hook](#)

Definition at line 74 of file ng_bt3c_var.h.

Referenced by bt3c_forward(), ng_bt3c_connect(), ng_bt3c_disconnect(), ng_bt3c_newhook(), ng_bt3c_rcvdata(), and ng_bt3c_rcvmsg().

6.2.2.5 [struct ifqueue bt3c_softc::inq](#)

Definition at line 98 of file ng_bt3c_var.h.

Referenced by bt3c_forward(), bt3c_pccard_attach(), bt3c_pccard_detach(), ng_bt3c_disconnect(), and ng_bt3c_rcvmsg().

6.2.2.6 [struct resource* bt3c_softc::iobase](#)

Definition at line 65 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.7 [int bt3c_softc::iobase_rid](#)

Definition at line 64 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.8 bus_space_handle_t bt3c_softc::ioh

Definition at line 67 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach().

6.2.2.9 bus_space_tag_t bt3c_softc::iot

Definition at line 66 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach().

6.2.2.10 struct resource* bt3c_softc::irq

Definition at line 69 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.11 void* bt3c_softc::irq_cookie

Definition at line 70 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.12 int bt3c_softc::irq_rid

Definition at line 68 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.13 void* bt3c_softc::ith

Definition at line 93 of file ng_bt3c_var.h.

Referenced by bt3c_intr(), bt3c_pccard_attach(), and bt3c_pccard_detach().

6.2.2.14 struct mbuf* bt3c_softc::m

Definition at line 95 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_detach(), and bt3c_receive().

6.2.2.15 node_p bt3c_softc::node

Definition at line 73 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach(), bt3c_pccard_detach(), bt3c_swi_intr(), ng_bt3c_rcvdata(), and ng_bt3c_shutdown().

6.2.2.16 struct ifqueue bt3c_softc::outq

Definition at line 99 of file ng_bt3c_var.h.

Referenced by `bt3c_pccard_attach()`, `bt3c_pccard_detach()`, `bt3c_send()`, `ng_bt3c_disconnect()`, `ng_bt3c_rcvdata()`, and `ng_bt3c_rcvmsg()`.

6.2.2.17 `ng_bt3c_node_stat_ep bt3c_softc::stat`

Definition at line 83 of file `ng_bt3c_var.h`.

Referenced by `bt3c_forward()`, `bt3c_receive()`, `bt3c_send()`, `ng_bt3c_rcvdata()`, and `ng_bt3c_rcvmsg()`.

6.2.2.18 `ng_bt3c_node_state_ep bt3c_softc::state`

Definition at line 81 of file `ng_bt3c_var.h`.

Referenced by `bt3c_pccard_attach()`, `bt3c_receive()`, and `ng_bt3c_rcvmsg()`.

6.2.2.19 `u_int32_t bt3c_softc::status`

Definition at line 92 of file `ng_bt3c_var.h`.

Referenced by `bt3c_swi_intr()`.

6.2.2.20 `u_int32_t bt3c_softc::want`

Definition at line 96 of file `ng_bt3c_var.h`.

Referenced by `bt3c_pccard_attach()`, and `bt3c_receive()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h`

6.3 ccatm_op Struct Reference

```
#include <ng_ccatm.h>
```

Data Fields

- uint32_t [op](#)
- u_char [data](#) []

6.3.1 Detailed Description

Definition at line 167 of file `ng_ccatm.h`.

6.3.2 Field Documentation

6.3.2.1 u_char [ccatm_op::data](#) []

Definition at line 169 of file `ng_ccatm.h`.

6.3.2.2 uint32_t [ccatm_op::op](#)

Definition at line 168 of file `ng_ccatm.h`.

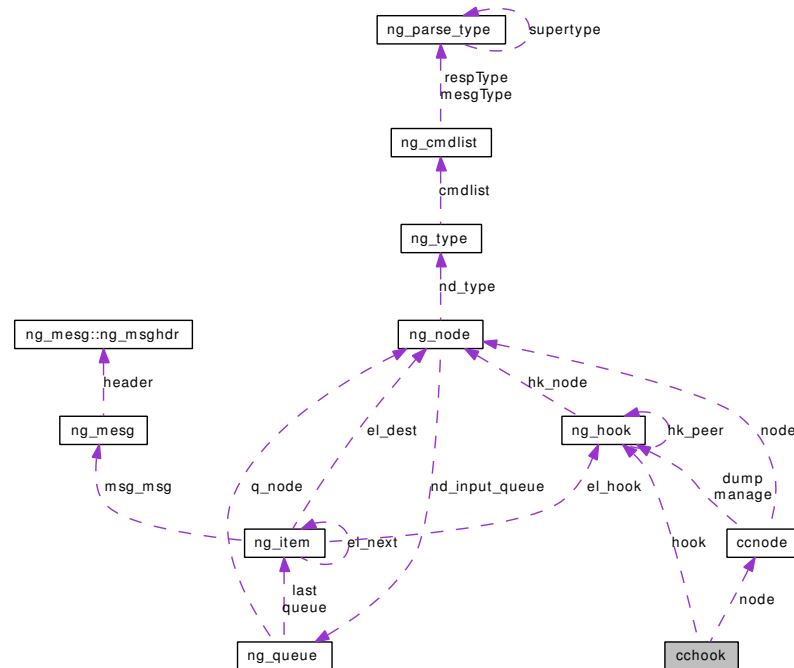
Referenced by `ng_ccatm_rcvdata()`, `ng_ccatm_respond_user()`, and `ng_ccatm_send_user()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_ccatm.h`

6.4 chook Struct Reference

Collaboration diagram for cchook:



Data Fields

- int `is_uni`
- `ccnode * node`
- `hook_p hook`
- void * `inst`

6.4.1 Detailed Description

Definition at line 318 of file `ng_ccatm.c`.

6.4.2 Field Documentation

6.4.2.1 `hook_p cchook::hook`

Definition at line 321 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_newhook()`, `ng_ccatm_rcvdata()`, `ng_ccatm_rcvuni()`, `ng_ccatm_respond_user()`, `ng_ccatm_send_uni()`, `ng_ccatm_send_uni_glob()`, and `ng_ccatm_send_user()`.

6.4.2.2 void* `cchook::inst`

Definition at line 322 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_rcvdata()`, and `ng_ccatm_rcvuni()`.

6.4.2.3 `int chook::is_uni`

Definition at line 319 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`.

6.4.2.4 `struct ccnode* chook::node`

Definition at line 320 of file `ng_ccatm.c`.

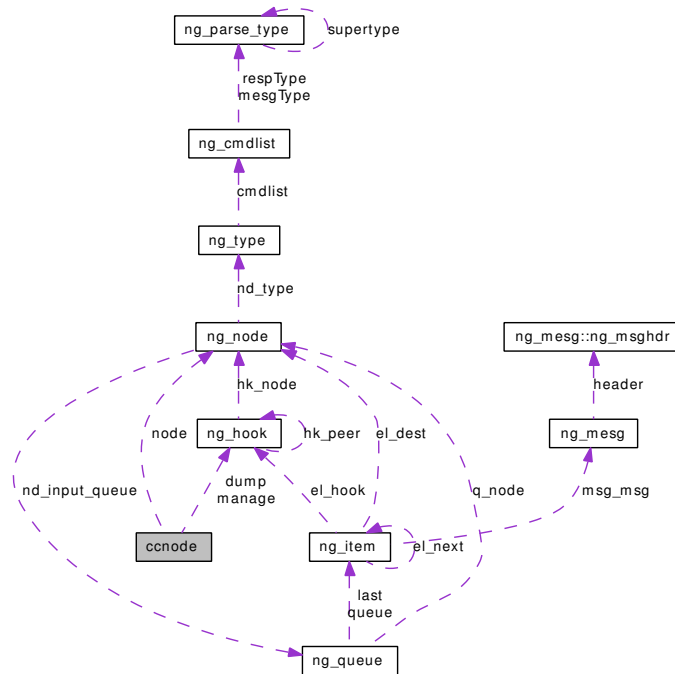
Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_rcvdata()`, and `ng_ccatm_rcvuni()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c](#)

6.5 cnode Struct Reference

Collaboration diagram for cnode:



Data Fields

- `node_p` `node`
- `hook_p` `dump`
- `hook_p` `manage`
- `ccdata` * `data`
- `mbuf` * `dump_first`
- `mbuf` * `dump_last`
- `u_int` `hook_cnt`

6.5.1 Detailed Description

Definition at line 303 of file `ng_ccatm.c`.

6.5.2 Field Documentation

6.5.2.1 struct `ccdata*` `ccnode::data`

Definition at line 308 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_dump()`, `ng_ccatm_get_addresses()`, `ng_ccatm_newhook()`, `ng_ccatm_rcvdata()`, `ng_ccatm_rcvmsg()`, `ng_ccatm_rcvuni()`, and `ng_ccatm_shutdown()`.

6.5.2.2 `hook_p cnode::dump`

Definition at line 305 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_dump()`, `ng_ccatm_newhook()`, `ng_ccatm_rcvmsg()`, and `send_dump()`.

6.5.2.3 `struct mbuf* cnode::dump_first`

Definition at line 309 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_dump()`, and `send_dump()`.

6.5.2.4 `struct mbuf* cnode::dump_last`

Definition at line 310 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_dump()`, and `send_dump()`.

6.5.2.5 `u_int cnode::hook_cnt`

Definition at line 312 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, `ng_ccatm_newhook()`, and `ng_ccatm_rcvmsg()`.

6.5.2.6 `hook_p cnode::manage`

Definition at line 306 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_disconnect()`, and `ng_ccatm_newhook()`.

6.5.2.7 `node_p cnode::node`

Definition at line 304 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_constructor()`, `ng_ccatm_disconnect()`, `ng_ccatm_dump()`, `ng_ccatm_get_addresses()`, `ng_ccatm_newhook()`, `ng_ccatm_rcvmsg()`, and `ng_ccatm_shutdown()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c`

6.6 cisco_header Struct Reference

Data Fields

- u_char [address](#)
- u_char [control](#)
- u_short [protocol](#)

6.6.1 Detailed Description

Definition at line 77 of file `ng_cisco.c`.

6.6.2 Field Documentation

6.6.2.1 u_char `cisco_header::address`

Definition at line 78 of file `ng_cisco.c`.

Referenced by `cisco_rcvdata()`, and `cisco_send()`.

6.6.2.2 u_char `cisco_header::control`

Definition at line 79 of file `ng_cisco.c`.

Referenced by `cisco_rcvdata()`, and `cisco_send()`.

6.6.2.3 u_short `cisco_header::protocol`

Definition at line 80 of file `ng_cisco.c`.

Referenced by `cisco_rcvdata()`, and `cisco_send()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_cisco.c`

6.7 cisco_packet Struct Reference

Data Fields

- u_long [type](#)
- u_long [par1](#)
- u_long [par2](#)
- u_short [rel](#)
- u_short [time0](#)
- u_short [time1](#)

6.7.1 Detailed Description

Definition at line 85 of file ng_cisco.c.

6.7.2 Field Documentation

6.7.2.1 u_long [cisco_packet::par1](#)

Definition at line 87 of file ng_cisco.c.

Referenced by [cisco_send\(\)](#).

6.7.2.2 u_long [cisco_packet::par2](#)

Definition at line 88 of file ng_cisco.c.

Referenced by [cisco_send\(\)](#).

6.7.2.3 u_short [cisco_packet::rel](#)

Definition at line 89 of file ng_cisco.c.

Referenced by [cisco_send\(\)](#).

6.7.2.4 u_short [cisco_packet::time0](#)

Definition at line 90 of file ng_cisco.c.

Referenced by [cisco_send\(\)](#).

6.7.2.5 u_short [cisco_packet::time1](#)

Definition at line 91 of file ng_cisco.c.

Referenced by [cisco_send\(\)](#).

6.7.2.6 u_long cisco_packet::type

Definition at line 86 of file `ng_cisco.c`.

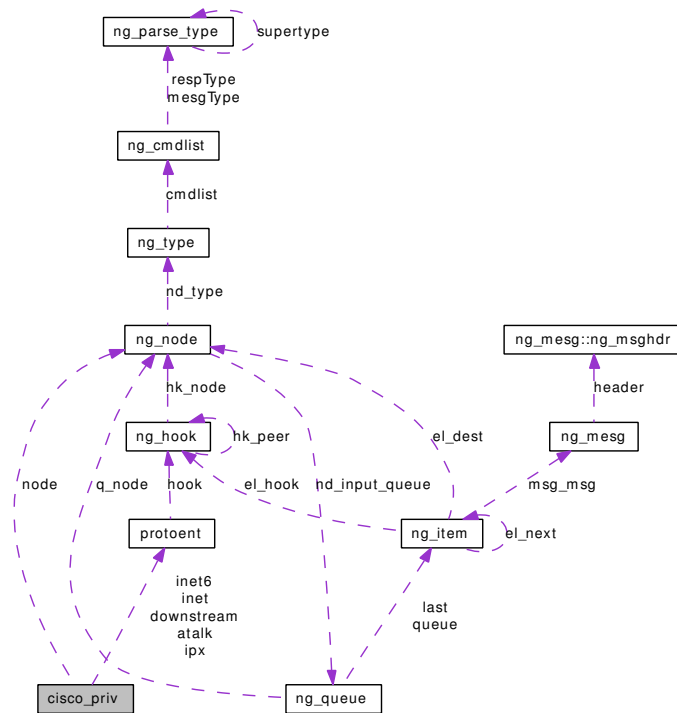
Referenced by `cisco_send()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_cisco.c`

6.8 cisco_priv Struct Reference

Collaboration diagram for cisco_priv:



Data Fields

- u_long [local_seq](#)
- u_long [remote_seq](#)
- u_long [seqRetries](#)
- node_p [node](#)
- callout [handle](#)
- protoent [downstream](#)
- protoent [inet](#)
- in_addr [localip](#)
- in_addr [localmask](#)
- protoent [inet6](#)
- protoent [atalk](#)
- protoent [ipx](#)

6.8.1 Detailed Description

Definition at line 101 of file `ng_cisco.c`.

6.8.2 Field Documentation

6.8.2.1 struct protoent cisco_priv::atalk

Definition at line 113 of file ng_cisco.c.

6.8.2.2 struct protoent cisco_priv::downstream

Definition at line 108 of file ng_cisco.c.

6.8.2.3 struct callout cisco_priv::handle

Definition at line 107 of file ng_cisco.c.

6.8.2.4 struct protoent cisco_priv::inet

Definition at line 109 of file ng_cisco.c.

6.8.2.5 struct protoent cisco_priv::inet6

Definition at line 112 of file ng_cisco.c.

6.8.2.6 struct protoent cisco_priv::ipx

Definition at line 114 of file ng_cisco.c.

6.8.2.7 u_long cisco_priv::local_seq

Definition at line 102 of file ng_cisco.c.

6.8.2.8 struct in_addr cisco_priv::localip

Definition at line 110 of file ng_cisco.c.

6.8.2.9 struct in_addr cisco_priv::localmask

Definition at line 111 of file ng_cisco.c.

6.8.2.10 node_p cisco_priv::node

Definition at line 106 of file ng_cisco.c.

6.8.2.11 u_long cisco_priv::remote_seq

Definition at line 103 of file ng_cisco.c.

6.8.2.12 u_long [cisco_priv::seqRetries](#)

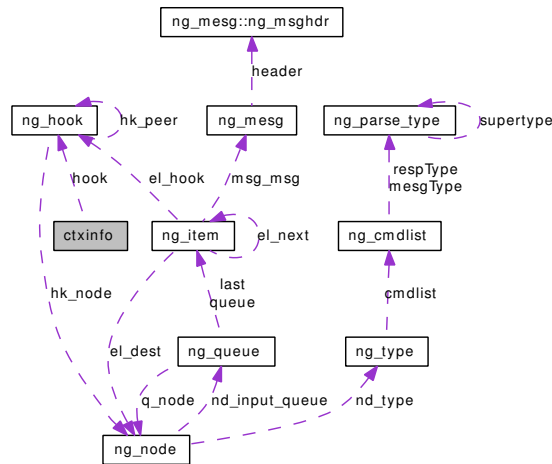
Definition at line 104 of file [ng_cisco.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_cisco.c](#)

6.9 ctxinfo Struct Reference

Collaboration diagram for ctxinfo:



Data Fields

- [u_int flags](#)
- [int dlcI](#)
- [hook_p hook](#)

6.9.1 Detailed Description

Definition at line 69 of file ng_frame_relay.c.

6.9.2 Field Documentation

6.9.2.1 int [ctxinfo::dlci](#)

Definition at line 73 of file ng_frame_relay.c.

Referenced by [ngfrm_disconnect\(\)](#), [ngfrm_newhook\(\)](#), and [ngfrm_rcvdata\(\)](#).

6.9.2.2 u_int [ctxinfo::flags](#)

Definition at line 70 of file ng_frame_relay.c.

Referenced by [ngfrm_disconnect\(\)](#), and [ngfrm_rcvdata\(\)](#).

6.9.2.3 [hook_p ctxinfo::hook](#)

Definition at line 74 of file ng_frame_relay.c.

Referenced by [ngfrm_disconnect\(\)](#), [ngfrm_newhook\(\)](#), and [ngfrm_rcvdata\(\)](#).

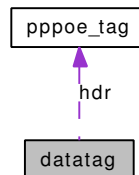
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_frame_relay.c](#)

6.10 datatag Struct Reference

```
#include <ng_pppoe.h>
```

Collaboration diagram for datatag:



Data Fields

- [pppoe_tag hdr](#)
- [u_int8_t data](#) [PPPOE_SERVICE_NAME_SIZE]

6.10.1 Detailed Description

Definition at line 232 of file `ng_pppoe.h`.

6.10.2 Field Documentation

6.10.2.1 `u_int8_t datatag::data`[PPPOE_SERVICE_NAME_SIZE]

Definition at line 234 of file `ng_pppoe.h`.

Referenced by `ng_pppoe_rcvmsg()`, `pppoe_find_svc()`, and `pppoe_match_svc()`.

6.10.2.2 `struct pppoe_tag datatag::hdr`

Definition at line 233 of file `ng_pppoe.h`.

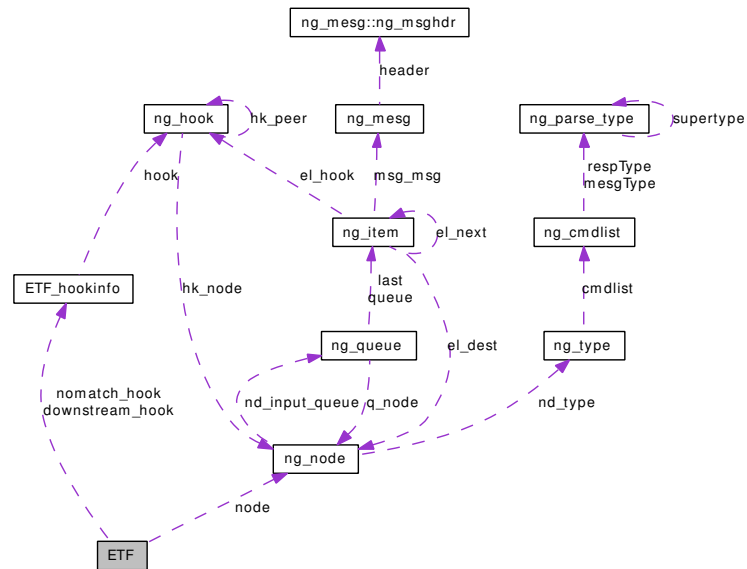
Referenced by `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, and `pppoe_start()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_pppoe.h`

6.11 ETF Struct Reference

Collaboration diagram for ETF:



Data Fields

- [ETF_hookinfo](#) [downstream_hook](#)
- [ETF_hookinfo](#) [nomatch_hook](#)
- [node_p](#) [node](#)
- [u_int](#) [packets_in](#)
- [u_int](#) [packets_out](#)
- [u_int32_t](#) [flags](#)
- [filterhead](#) [hashtable](#) [HASHSIZE]

6.11.1 Detailed Description

Definition at line 144 of file `ng_etf.c`.

6.11.2 Field Documentation

6.11.2.1 struct [ETF_hookinfo](#) [ETF::downstream_hook](#)

Definition at line 145 of file `ng_etf.c`.

Referenced by `ng_etf_newhook()`, `ng_etf_rcvdata()`, and `ng_etf_rcvmsg()`.

6.11.2.2 [u_int32_t](#) [ETF::flags](#)

Definition at line 150 of file `ng_etf.c`.

Referenced by `ng_etf_rcvmsg()`.

6.11.2.3 struct filterhead [ETF::hashtable](#)[HASHSIZE]

Definition at line 151 of file `ng_etf.c`.

Referenced by `ng_etf_disconnect()`, `ng_etf_findentry()`, and `ng_etf_rcvmsg()`.

6.11.2.4 [node_p](#) [ETF::node](#)

Definition at line 147 of file `ng_etf.c`.

Referenced by `ng_etf_shutdown()`.

6.11.2.5 struct [ETF_hookinfo](#) [ETF::nomatch_hook](#)

Definition at line 146 of file `ng_etf.c`.

Referenced by `ng_etf_newhook()`, and `ng_etf_rcvdata()`.

6.11.2.6 [u_int](#) [ETF::packets_in](#)

Definition at line 148 of file `ng_etf.c`.

Referenced by `ng_etf_newhook()`, `ng_etf_rcvdata()`, and `ng_etf_rcvmsg()`.

6.11.2.7 [u_int](#) [ETF::packets_out](#)

Definition at line 149 of file `ng_etf.c`.

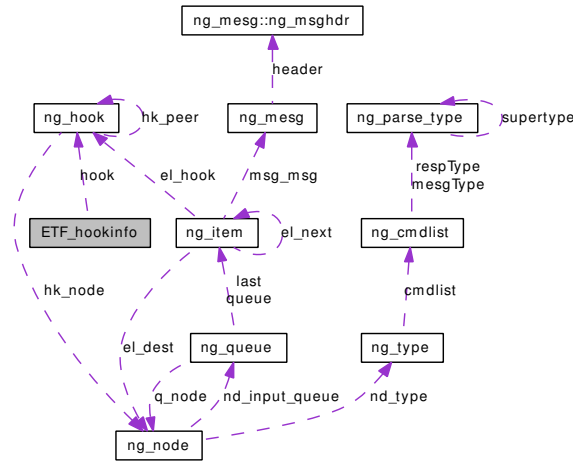
Referenced by `ng_etf_newhook()`, `ng_etf_rcvdata()`, and `ng_etf_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_etf.c`

6.12 ETF_hookinfo Struct Reference

Collaboration diagram for ETF_hookinfo:



Data Fields

- [hook_p hook](#)

6.12.1 Detailed Description

Definition at line 129 of file `ng_etf.c`.

6.12.2 Field Documentation

6.12.2.1 [hook_p ETF_hookinfo::hook](#)

Definition at line 130 of file `ng_etf.c`.

Referenced by `ng_etf_newhook()`, `ng_etf_rcvdata()`, and `ng_etf_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_etf.c`

6.13 filter Struct Reference

6.13.1 Detailed Description

Definition at line 133 of file `ng_etf.c`.

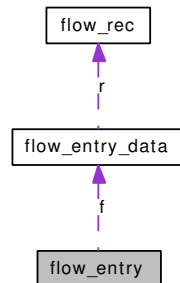
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_etf.c](#)

6.14 flow_entry Struct Reference

```
#include <ng_netflow.h>
```

Collaboration diagram for flow_entry:



Public Member Functions

- [TAILQ_ENTRY \(flow_entry\)](#) fle_hash

Data Fields

- [flow_entry_data](#) f

6.14.1 Detailed Description

Definition at line 159 of file ng_netflow.h.

6.14.2 Member Function Documentation

6.14.2.1 flow_entry::TAILQ_ENTRY (flow_entry)

6.14.3 Field Documentation

6.14.3.1 struct flow_entry_data flow_entry::f

Definition at line 160 of file ng_netflow.h.

Referenced by export_add(), hash_insert(), and ng_netflow_flow_add().

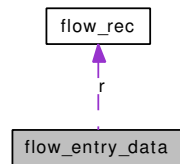
The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/netflow/ng_netflow.h

6.15 flow_entry_data Struct Reference

```
#include <ng_netflow.h>
```

Collaboration diagram for flow_entry_data:



Data Fields

- [flow_rec](#) `r`
- `in_addr` [next_hop](#)
- `uint16_t` [fle_o_ifx](#)
- `uint8_t` [dst_mask](#)
- `uint8_t` [src_mask](#)
- `u_long` [packets](#)
- `u_long` [bytes](#)
- `long` [first](#)
- `long` [last](#)
- `u_char` [tcp_flags](#)

6.15.1 Detailed Description

Definition at line 125 of file `ng_netflow.h`.

6.15.2 Field Documentation

6.15.2.1 `u_long` [flow_entry_data::bytes](#)

Definition at line 133 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ng_netflow_flow_add()`.

6.15.2.2 `uint8_t` [flow_entry_data::dst_mask](#)

Definition at line 130 of file `ng_netflow.h`.

Referenced by `export_add()`, and `hash_insert()`.

6.15.2.3 `long` [flow_entry_data::first](#)

Definition at line 134 of file `ng_netflow.h`.

Referenced by `export_add()`, and `hash_insert()`.

6.15.2.4 `uint16_t flow_entry_data::fle_o_ifx`

Definition at line 128 of file `ng_netflow.h`.

Referenced by `export_add()`, and `hash_insert()`.

6.15.2.5 `long flow_entry_data::last`

Definition at line 135 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ng_netflow_flow_add()`.

6.15.2.6 `struct in_addr flow_entry_data::next_hop`

Definition at line 127 of file `ng_netflow.h`.

Referenced by `export_add()`, and `hash_insert()`.

6.15.2.7 `u_long flow_entry_data::packets`

Definition at line 132 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ng_netflow_flow_add()`.

6.15.2.8 `struct flow_rec flow_entry_data::r`

Definition at line 126 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ng_netflow_flow_add()`.

6.15.2.9 `uint8_t flow_entry_data::src_mask`

Definition at line 131 of file `ng_netflow.h`.

Referenced by `export_add()`, and `hash_insert()`.

6.15.2.10 `u_char flow_entry_data::tcp_flags`

Definition at line 136 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ng_netflow_flow_add()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.16 flow_hash_entry Struct Reference

```
#include <ng_netflow.h>
```

Public Member Functions

- [TAILQ_HEAD](#) (fhead, [flow_entry](#)) head

Data Fields

- [mtx](#) [mtx](#)

6.16.1 Detailed Description

Definition at line 259 of file [ng_netflow.h](#).

6.16.2 Member Function Documentation

6.16.2.1 [flow_hash_entry::TAILQ_HEAD](#) (fhead, [flow_entry](#))

6.16.3 Field Documentation

6.16.3.1 [struct](#) [mtx](#) [flow_hash_entry::mtx](#)

Definition at line 260 of file [ng_netflow.h](#).

Referenced by [hash_insert\(\)](#), [ng_netflow_cache_flush\(\)](#), [ng_netflow_cache_init\(\)](#), [ng_netflow_expire\(\)](#), [ng_netflow_flow_add\(\)](#), and [ng_netflow_flow_show\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/ng_netflow.h](#)

6.17 flow_manager Struct Reference

```
#include <ng_message.h>
```

Data Fields

- [ng_ID_t id](#)

6.17.1 Detailed Description

Definition at line 360 of file `ng_message.h`.

6.17.2 Field Documentation

6.17.2.1 `ng_ID_t flow_manager::id`

Definition at line 361 of file `ng_message.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.18 flow_rec Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- in_addr [r_src](#)
 - in_addr [r_dst](#)
 - union {
 - struct {
 - [uint16_t s_port](#)
 - [uint16_t d_port](#)
 - [} dir](#)
 - [uint32_t both](#)
 - [} ports](#)
-
- union {
 - struct {
 - [u_char prot](#)
 - [u_char tos](#)
 - [uint16_t i_ifx](#)
 - [} i](#)
 - [uint32_t all](#)
 - [} misc](#)

6.18.1 Detailed Description

Definition at line 96 of file `ng_netflow.h`.

6.18.2 Field Documentation

6.18.2.1 [uint32_t flow_rec::all](#)

Definition at line 112 of file `ng_netflow.h`.

6.18.2.2 [uint32_t flow_rec::both](#)

Definition at line 104 of file `ng_netflow.h`.

6.18.2.3 [uint16_t flow_rec::d_port](#)

Definition at line 102 of file `ng_netflow.h`.

6.18.2.4 `struct { ... } flow_rec::dir`

6.18.2.5 `struct { ... } flow_rec::i`

6.18.2.6 `uint16_t flow_rec::i_ifx`

Definition at line 110 of file `ng_netflow.h`.

6.18.2.7 `union { ... } flow_rec::misc`

6.18.2.8 `union { ... } flow_rec::ports`

6.18.2.9 `u_char flow_rec::prot`

Definition at line 108 of file `ng_netflow.h`.

6.18.2.10 `struct in_addr flow_rec::r_dst`

Definition at line 98 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ip_hash()`.

6.18.2.11 `struct in_addr flow_rec::r_src`

Definition at line 97 of file `ng_netflow.h`.

Referenced by `export_add()`, `hash_insert()`, and `ip_hash()`.

6.18.2.12 `uint16_t flow_rec::s_port`

Definition at line 101 of file `ng_netflow.h`.

6.18.2.13 `u_char flow_rec::tos`

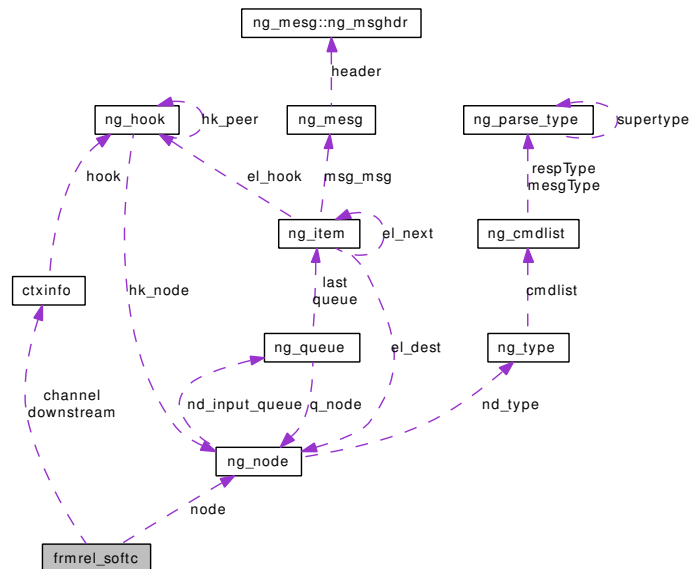
Definition at line 109 of file `ng_netflow.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.19 frmrel_softc Struct Reference

Collaboration diagram for frmrel_softc:



Data Fields

- int [unit](#)
- int [datahooks](#)
- [node_p](#) node
- int [addrlen](#)
- int [flags](#)
- int [mtu](#)
- u_char [remote_seq](#)
- u_char [local_seq](#)
- u_short [ALT](#) [1024]
- [ctxinfo](#) [channel](#) [MAX_CT]
- [ctxinfo](#) [downstream](#)

6.19.1 Detailed Description

Definition at line 78 of file `ng_frame_relay.c`.

6.19.2 Field Documentation

6.19.2.1 int `frmrel_softc::addrlen`

Definition at line 82 of file `ng_frame_relay.c`.

6.19.2.2 `u_short frmrel_softc::ALT[1024]`

Definition at line 87 of file `ng_frame_relay.c`.

6.19.2.3 `struct ctxinfo frmrel_softc::channel[MAX_CT]`

Definition at line 90 of file `ng_frame_relay.c`.

6.19.2.4 `int frmrel_softc::datahooks`

Definition at line 80 of file `ng_frame_relay.c`.

6.19.2.5 `struct ctxinfo frmrel_softc::downstream`

Definition at line 91 of file `ng_frame_relay.c`.

6.19.2.6 `int frmrel_softc::flags`

Definition at line 83 of file `ng_frame_relay.c`.

6.19.2.7 `u_char frmrel_softc::local_seq`

Definition at line 86 of file `ng_frame_relay.c`.

6.19.2.8 `int frmrel_softc::mtu`

Definition at line 84 of file `ng_frame_relay.c`.

6.19.2.9 `node_p frmrel_softc::node`

Definition at line 81 of file `ng_frame_relay.c`.

6.19.2.10 `u_char frmrel_softc::remote_seq`

Definition at line 85 of file `ng_frame_relay.c`.

6.19.2.11 `int frmrel_softc::unit`

Definition at line 79 of file `ng_frame_relay.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_frame_relay.c`

6.20 greheader Struct Reference

Data Fields

- u_char [recursion](#):3
- u_char [ssr](#):1
- u_char [hasSeq](#):1
- u_char [hasKey](#):1
- u_char [hasRoute](#):1
- u_char [hasSum](#):1
- u_char [vers](#):3
- u_char [flags](#):4
- u_char [hasAck](#):1
- u_int16_t [proto](#)
- u_int16_t [length](#)
- u_int16_t [cid](#)
- u_int32_t [data](#) [0]

6.20.1 Detailed Description

Definition at line 77 of file ng_pptpgre.c.

6.20.2 Field Documentation

6.20.2.1 u_int16_t [greheader::cid](#)

Definition at line 103 of file ng_pptpgre.c.

Referenced by [ng_pptpgre_xmit\(\)](#).

6.20.2.2 u_int32_t [greheader::data\[0\]](#)

Definition at line 104 of file ng_pptpgre.c.

Referenced by [ng_pptpgre_xmit\(\)](#).

6.20.2.3 u_char [greheader::flags](#)

Definition at line 86 of file ng_pptpgre.c.

6.20.2.4 u_char [greheader::hasAck](#)

Definition at line 87 of file ng_pptpgre.c.

Referenced by [ng_pptpgre_xmit\(\)](#).

6.20.2.5 u_char [greheader::hasKey](#)

Definition at line 82 of file ng_pptpgre.c.

6.20.2.6 `u_char greheader::hasRoute`

Definition at line 83 of file `ng_pptpgre.c`.

6.20.2.7 `u_char greheader::hasSeq`

Definition at line 81 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_xmit()`.

6.20.2.8 `u_char greheader::hasSum`

Definition at line 84 of file `ng_pptpgre.c`.

6.20.2.9 `u_int16_t greheader::length`

Definition at line 102 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_xmit()`.

6.20.2.10 `u_int16_t greheader::proto`

Definition at line 101 of file `ng_pptpgre.c`.

6.20.2.11 `u_char greheader::recursion`

Definition at line 79 of file `ng_pptpgre.c`.

6.20.2.12 `u_char greheader::ssr`

Definition at line 80 of file `ng_pptpgre.c`.

6.20.2.13 `u_char greheader::vers`

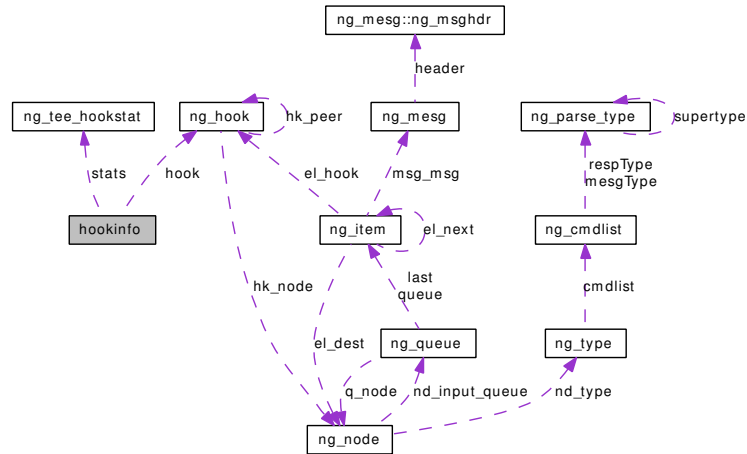
Definition at line 85 of file `ng_pptpgre.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_pptpgre.c`

6.21 hookinfo Struct Reference

Collaboration diagram for hookinfo:



Data Fields

- [hook_p](#) hook
- [ng_tee_hookstat](#) stats

6.21.1 Detailed Description

Definition at line 66 of file `ng_tee.c`.

6.21.2 Field Documentation

6.21.2.1 [hook_p](#) `hookinfo::hook`

Definition at line 67 of file `ng_tee.c`.

Referenced by `ngt_disconnect()`, and `ngt_rcvdata()`.

6.21.2.2 `struct ng_tee_hookstat` `hookinfo::stats`

Definition at line 68 of file `ng_tee.c`.

Referenced by `ngt_rcvdata()`.

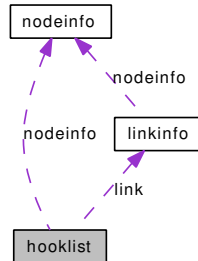
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_tee.c`

6.22 hooklist Struct Reference

```
#include <ng_message.h>
```

Collaboration diagram for hooklist:



Data Fields

- [nodeinfo](#) [nodeinfo](#)
- [linkinfo](#) [link](#) []

6.22.1 Detailed Description

Definition at line 264 of file `ng_message.h`.

6.22.2 Field Documentation

6.22.2.1 struct [linkinfo](#) `hooklist::link` []

Definition at line 266 of file `ng_message.h`.

6.22.2.2 struct [nodeinfo](#) `hooklist::nodeinfo`

Definition at line 265 of file `ng_message.h`.

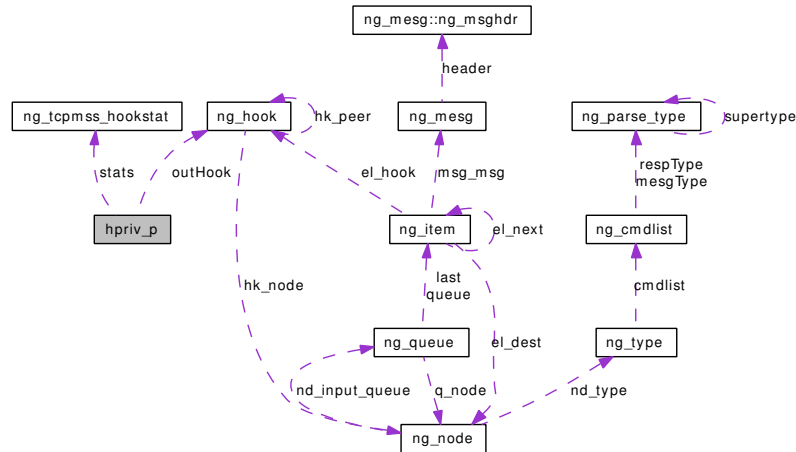
Referenced by `ng_generic_linkinfo_getLength()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.23 hpriv_p Struct Reference

Collaboration diagram for hpriv_p:



Data Fields

- [hook_p outHook](#)
- [ng_tcpmss_hookstat stats](#)

6.23.1 Detailed Description

Definition at line 66 of file `ng_tcpmss.c`.

6.23.2 Field Documentation

6.23.2.1 [hook_p hpriv_p::outHook](#)

Definition at line 67 of file `ng_tcpmss.c`.

Referenced by `ng_tcpmss_disconnect()`, `ng_tcpmss_rcvdata()`, and `ng_tcpmss_rcvmsg()`.

6.23.2.2 **struct [ng_tcpmss_hookstat](#) `hpriv_p::stats`**

Definition at line 68 of file `ng_tcpmss.c`.

Referenced by `ng_tcpmss_rcvdata()`, and `ng_tcpmss_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_tcpmss.c`

6.24 hva_stats_aal5 Struct Reference

```
#include <ng_atmpif.h>
```

Data Fields

- [uint64_t aal5_xmit](#)
- [uint64_t aal5_rcvd](#)
- [uint32_t aal5_crc_len](#)
- [uint32_t aal5_drops](#)
- [uint64_t aal5_pdu_xmit](#)
- [uint64_t aal5_pdu_rcvd](#)
- [uint32_t aal5_pdu_crc](#)
- [uint32_t aal5_pdu_errs](#)
- [uint32_t aal5_pdu_drops](#)

6.24.1 Detailed Description

Definition at line 113 of file `ng_atmpif.h`.

6.24.2 Field Documentation

6.24.2.1 [uint32_t hva_stats_aal5::aal5_crc_len](#)

Definition at line 116 of file `ng_atmpif.h`.

6.24.2.2 [uint32_t hva_stats_aal5::aal5_drops](#)

Definition at line 117 of file `ng_atmpif.h`.

6.24.2.3 [uint32_t hva_stats_aal5::aal5_pdu_crc](#)

Definition at line 120 of file `ng_atmpif.h`.

6.24.2.4 [uint32_t hva_stats_aal5::aal5_pdu_drops](#)

Definition at line 122 of file `ng_atmpif.h`.

6.24.2.5 [uint32_t hva_stats_aal5::aal5_pdu_errs](#)

Definition at line 121 of file `ng_atmpif.h`.

6.24.2.6 [uint64_t hva_stats_aal5::aal5_pdu_rcvd](#)

Definition at line 119 of file `ng_atmpif.h`.

6.24.2.7 `uint64_t` [hva_stats_aal5::aal5_pdu_xmit](#)

Definition at line 118 of file `ng_atmpif.h`.

6.24.2.8 `uint64_t` [hva_stats_aal5::aal5_rcvd](#)

Definition at line 115 of file `ng_atmpif.h`.

6.24.2.9 `uint64_t` [hva_stats_aal5::aal5_xmit](#)

Definition at line 114 of file `ng_atmpif.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atmpif.h`

6.25 hva_stats_atm Struct Reference

```
#include <ng_atmpif.h>
```

Data Fields

- `uint64_t atm_xmit`
- `uint64_t atm_rcvd`

6.25.1 Detailed Description

Definition at line 103 of file `ng_atmpif.h`.

6.25.2 Field Documentation

6.25.2.1 `uint64_t hva_stats_atm::atm_rcvd`

Definition at line 105 of file `ng_atmpif.h`.

6.25.2.2 `uint64_t hva_stats_atm::atm_xmit`

Definition at line 104 of file `ng_atmpif.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atmpif.h`

6.26 hva_stats_ng Struct Reference

```
#include <ng_atmpif.h>
```

Data Fields

- [uint32_t ng_errseq](#)
- [uint32_t ng_lostpdu](#)
- [uint32_t ng_badpdu](#)
- [uint32_t ng_rx_novcc](#)
- [uint32_t ng_rx_iqfull](#)
- [uint32_t ng_tx_rawcell](#)
- [uint32_t ng_rx_rawcell](#)
- [uint64_t ng_tx_pdu](#)
- [uint64_t ng_rx_pdu](#)

6.26.1 Detailed Description

Definition at line 78 of file ng_atmpif.h.

6.26.2 Field Documentation

6.26.2.1 [uint32_t hva_stats_ng::ng_badpdu](#)

Definition at line 81 of file ng_atmpif.h.

6.26.2.2 [uint32_t hva_stats_ng::ng_errseq](#)

Definition at line 79 of file ng_atmpif.h.

6.26.2.3 [uint32_t hva_stats_ng::ng_lostpdu](#)

Definition at line 80 of file ng_atmpif.h.

6.26.2.4 [uint32_t hva_stats_ng::ng_rx_iqfull](#)

Definition at line 83 of file ng_atmpif.h.

6.26.2.5 [uint32_t hva_stats_ng::ng_rx_novcc](#)

Definition at line 82 of file ng_atmpif.h.

6.26.2.6 [uint64_t hva_stats_ng::ng_rx_pdu](#)

Definition at line 87 of file ng_atmpif.h.

6.26.2.7 `uint32_t hva_stats_ng::ng_rx_rawcell`

Definition at line 85 of file `ng_atmpif.h`.

6.26.2.8 `uint64_t hva_stats_ng::ng_tx_pdu`

Definition at line 86 of file `ng_atmpif.h`.

6.26.2.9 `uint32_t hva_stats_ng::ng_tx_rawcell`

Definition at line 84 of file `ng_atmpif.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atmpif.h`

6.27 iffam Struct Reference

Data Fields

- `sa_family_t` [family](#)
- `const char *` [hookname](#)
- `const char *` [hookname](#)

6.27.1 Detailed Description

Definition at line 99 of file `ng_gif_demux.c`.

6.27.2 Field Documentation

6.27.2.1 `sa_family_t iffam::family`

Definition at line 100 of file `ng_gif_demux.c`.

Referenced by `get_iffam_from_af()`, `ng_gif_demux_rcvdata()`, and `ng_iface_rcvdata()`.

6.27.2.2 `const char* iffam::hookname`

Definition at line 95 of file `ng_iface.c`.

6.27.2.3 `const char* iffam::hookname`

Definition at line 101 of file `ng_gif_demux.c`.

Referenced by `get_iffam_from_name()`.

The documentation for this struct was generated from the following files:

- `/usr/src/sys/netgraph/ng_gif_demux.c`
- `/usr/src/sys/netgraph/ng_iface.c`

6.28 int16_temp Struct Reference

Data Fields

- char [x](#)
- int16_t [y](#)

6.28.1 Detailed Description

Definition at line 71 of file [ng_parse.c](#).

6.28.2 Field Documentation

6.28.2.1 char [int16_temp::x](#)

Definition at line 72 of file [ng_parse.c](#).

6.28.2.2 int16_t [int16_temp::y](#)

Definition at line 73 of file [ng_parse.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.c](#)

6.29 int32_temp Struct Reference

Data Fields

- char [x](#)
- int32_t [y](#)

6.29.1 Detailed Description

Definition at line 76 of file [ng_parse.c](#).

6.29.2 Field Documentation

6.29.2.1 char [int32_temp::x](#)

Definition at line 77 of file [ng_parse.c](#).

6.29.2.2 int32_t [int32_temp::y](#)

Definition at line 78 of file [ng_parse.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.c](#)

6.30 int64_temp Struct Reference

Data Fields

- char [x](#)
- int64_t [y](#)

6.30.1 Detailed Description

Definition at line 81 of file [ng_parse.c](#).

6.30.2 Field Documentation

6.30.2.1 char [int64_temp::x](#)

Definition at line 82 of file [ng_parse.c](#).

6.30.2.2 int64_t [int64_temp::y](#)

Definition at line 83 of file [ng_parse.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.c](#)

6.31 l2tp_seq Struct Reference

Data Fields

- `u_int16_t ns`
- `u_int16_t nr`
- `u_int16_t rack`
- `u_int16_t xack`
- `u_int16_t wmax`
- `u_int16_t cwnd`
- `u_int16_t ssth`
- `u_int16_t acks`
- `u_int16_t rexmits`
- `u_int16_t max_rexmits`
- `u_int16_t max_rexmit_to`
- callout `rack_timer`
- callout `xack_timer`
- `mbuf * xwin` [L2TP_MAX_XWIN]

6.31.1 Detailed Description

Definition at line 118 of file `ng_l2tp.c`.

6.31.2 Field Documentation

6.31.2.1 `u_int16_t l2tp_seq::acks`

Definition at line 126 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_rack_timeout()`.

6.31.2.2 `u_int16_t l2tp_seq::cwnd`

Definition at line 124 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_ctrl()`, `ng_l2tp_seq_rack_timeout()`, and `ng_l2tp_seq_rcv_nr()`.

6.31.2.3 `u_int16_t l2tp_seq::max_rexmit_to`

Definition at line 129 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_adjust()`, and `ng_l2tp_seq_rack_timeout()`.

6.31.2.4 `u_int16_t l2tp_seq::max_rexmits`

Definition at line 128 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_adjust()`, and `ng_l2tp_seq_rack_timeout()`.

6.31.2.5 `u_int16_t l2tp_seq::nr`

Definition at line 120 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_rcv_ns()`, `ng_l2tp_seq_set()`, `ng_l2tp_seq_xack_timeout()`, and `ng_l2tp_xmit_ctrl()`.

6.31.2.6 `u_int16_t l2tp_seq::ns`

Definition at line 119 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_ctrl()`, `ng_l2tp_seq_rack_timeout()`, `ng_l2tp_seq_rcv_nr()`, `ng_l2tp_seq_rcv_ns()`, `ng_l2tp_seq_set()`, and `ng_l2tp_seq_xack_timeout()`.

6.31.2.7 `u_int16_t l2tp_seq::rack`

Definition at line 121 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_rack_timeout()`, `ng_l2tp_seq_rcv_nr()`, and `ng_l2tp_seq_set()`.

6.31.2.8 `struct callout l2tp_seq::rack_timer`

Definition at line 130 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_ctrl()`, `ng_l2tp_seq_rack_timeout()`, `ng_l2tp_seq_reset()`, and `ng_l2tp_shutdown()`.

6.31.2.9 `u_int16_t l2tp_seq::rexmits`

Definition at line 127 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_rack_timeout()`, and `ng_l2tp_seq_rcv_nr()`.

6.31.2.10 `u_int16_t l2tp_seq::sssth`

Definition at line 125 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_rack_timeout()`, and `ng_l2tp_seq_rcv_nr()`.

6.31.2.11 `u_int16_t l2tp_seq::wmax`

Definition at line 123 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_adjust()`, and `ng_l2tp_seq_rcv_nr()`.

6.31.2.12 `u_int16_t l2tp_seq::xack`

Definition at line 122 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_seq_set()`, `ng_l2tp_seq_xack_timeout()`, and `ng_l2tp_xmit_ctrl()`.

6.31.2.13 struct callout l2tp_seq::xack_timer

Definition at line 131 of file ng_l2tp.c.

Referenced by ng_l2tp_seq_recv_ns(), ng_l2tp_seq_reset(), ng_l2tp_shutdown(), and ng_l2tp_xmit_ctrl().

6.31.2.14 struct mbuf* l2tp_seq::xwin[L2TP_MAX_XWIN]

Definition at line 132 of file ng_l2tp.c.

Referenced by ng_l2tp_recv_ctrl(), ng_l2tp_seq_rack_timeout(), ng_l2tp_seq_recv_nr(), and ng_l2tp_seq_reset().

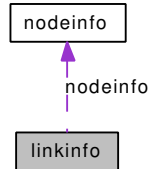
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_l2tp.c](#)

6.32 linkinfo Struct Reference

```
#include <ng_message.h>
```

Collaboration diagram for linkinfo:



Data Fields

- char [ourhook](#) [NG_HOOKSIZ]
- char [peerhook](#) [NG_HOOKSIZ]
- [nodeinfo](#) [nodeinfo](#)

6.32.1 Detailed Description

Definition at line 250 of file `ng_message.h`.

6.32.2 Field Documentation

6.32.2.1 struct [nodeinfo](#) [linkinfo::nodeinfo](#)

Definition at line 253 of file `ng_message.h`.

Referenced by `ng_generic_msg()`.

6.32.2.2 char [linkinfo::ourhook](#)[NG_HOOKSIZ]

Definition at line 251 of file `ng_message.h`.

Referenced by `ng_generic_msg()`.

6.32.2.3 char [linkinfo::peerhook](#)[NG_HOOKSIZ]

Definition at line 252 of file `ng_message.h`.

Referenced by `ng_generic_msg()`.

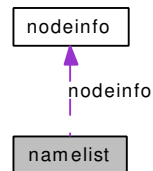
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.33 namelist Struct Reference

```
#include <ng_message.h>
```

Collaboration diagram for namelist:



Data Fields

- `u_int32_t numnames`
- `nodeinfo nodeinfo []`

6.33.1 Detailed Description

Definition at line 277 of file `ng_message.h`.

6.33.2 Field Documentation

6.33.2.1 struct `nodeinfo` `namelist::nodeinfo[]`

Definition at line 279 of file `ng_message.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

6.33.2.2 `u_int32_t` `namelist::numnames`

Definition at line 278 of file `ng_message.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

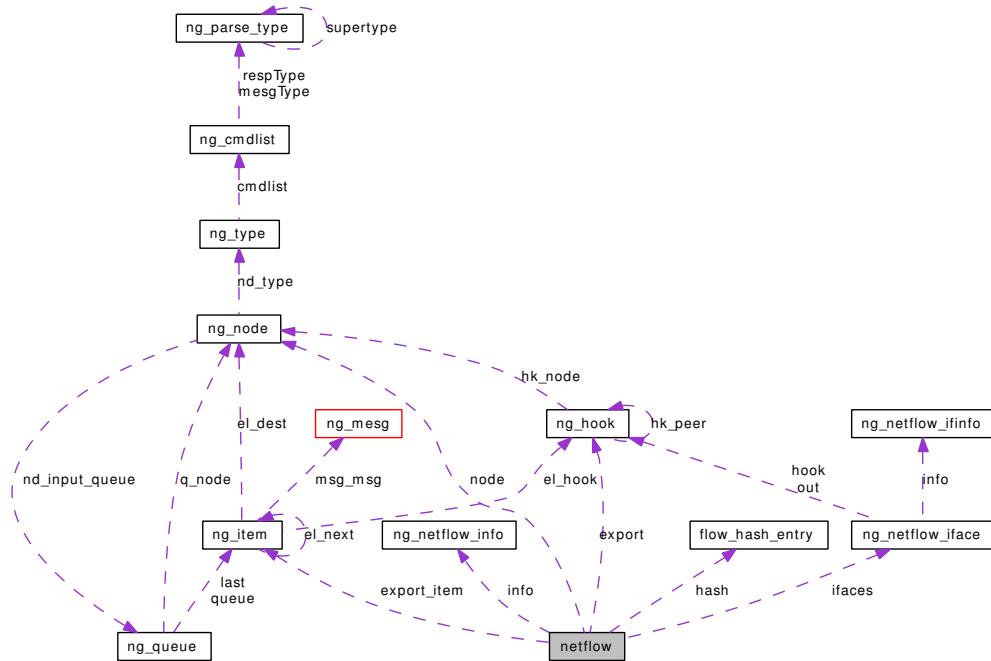
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.34 netflow Struct Reference

```
#include <ng_netflow.h>
```

Collaboration diagram for netflow:



Data Fields

- [node_p](#) node
- [hook_p](#) export
- [ng_netflow_info](#) info
- callout [exp_callout](#)
- [uma_zone_t](#) zone
- [flow_hash_entry](#) * hash
- [item_p](#) export_item
- [mtx](#) export_mtx
- [uint32_t](#) flow_seq
- [ng_netflow_iface](#) ifaces [NG_NETFLOW_MAXIFACES]

6.34.1 Detailed Description

Definition at line 220 of file ng_netflow.h.

6.34.2 Field Documentation

6.34.2.1 struct callout [netflow::exp_callout](#)

Definition at line 225 of file ng_netflow.h.

6.34.2.2 [hook_p netflow::export](#)

Definition at line 222 of file `ng_netflow.h`.

6.34.2.3 [item_p netflow::export_item](#)

Definition at line 249 of file `ng_netflow.h`.

6.34.2.4 [struct mtx netflow::export_mtx](#)

Definition at line 250 of file `ng_netflow.h`.

6.34.2.5 [uint32_t netflow::flow_seq](#)

Definition at line 251 of file `ng_netflow.h`.

6.34.2.6 [struct flow_hash_entry* netflow::hash](#)

Definition at line 236 of file `ng_netflow.h`.

6.34.2.7 [struct ng_netflow_iface netflow::ifaces\[NG_NETFLOW_MAXIFACES\]](#)

Definition at line 253 of file `ng_netflow.h`.

6.34.2.8 [struct ng_netflow_info netflow::info](#)

Definition at line 224 of file `ng_netflow.h`.

6.34.2.9 [node_p netflow::node](#)

Definition at line 221 of file `ng_netflow.h`.

6.34.2.10 [uma_zone_t netflow::zone](#)

Definition at line 235 of file `ng_netflow.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/ng_netflow.h](#)

6.35 netflow_v1_header Struct Reference

```
#include <netflow.h>
```

Data Fields

- [uint16_t version](#)
- [uint16_t count](#)
- [uint32_t sys_uptime](#)
- [uint32_t unix_secs](#)
- [uint32_t unix_nsecs](#)

6.35.1 Detailed Description

Definition at line 50 of file netflow.h.

6.35.2 Field Documentation

6.35.2.1 [uint16_t netflow_v1_header::count](#)

Definition at line 53 of file netflow.h.

6.35.2.2 [uint32_t netflow_v1_header::sys_uptime](#)

Definition at line 54 of file netflow.h.

6.35.2.3 [uint32_t netflow_v1_header::unix_nsecs](#)

Definition at line 56 of file netflow.h.

6.35.2.4 [uint32_t netflow_v1_header::unix_secs](#)

Definition at line 55 of file netflow.h.

6.35.2.5 [uint16_t netflow_v1_header::version](#)

Definition at line 52 of file netflow.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/netflow.h](#)

6.36 netflow_v1_record Struct Reference

```
#include <netflow.h>
```

Data Fields

- [uint32_t src_addr](#)
- [uint32_t dst_addr](#)
- [uint32_t next_hop](#)
- [uint16_t in_ifx](#)
- [uint16_t out_ifx](#)
- [uint32_t packets](#)
- [uint32_t octets](#)
- [uint32_t first](#)
- [uint32_t last](#)
- [uint16_t s_port](#)
- [uint16_t d_port](#)
- [uint16_t pad1](#)
- [uint8_t prot](#)
- [uint8_t tos](#)
- [uint8_t flags](#)
- [uint8_t pad2](#)
- [uint16_t pad3](#)
- [uint8_t reserved](#) [5]

6.36.1 Detailed Description

Definition at line 72 of file netflow.h.

6.36.2 Field Documentation

6.36.2.1 [uint16_t netflow_v1_record::d_port](#)

Definition at line 84 of file netflow.h.

6.36.2.2 [uint32_t netflow_v1_record::dst_addr](#)

Definition at line 75 of file netflow.h.

6.36.2.3 [uint32_t netflow_v1_record::first](#)

Definition at line 81 of file netflow.h.

6.36.2.4 [uint8_t netflow_v1_record::flags](#)

Definition at line 88 of file netflow.h.

6.36.2.5 `uint16_t netflow_v1_record::in_ifx`

Definition at line 77 of file netflow.h.

6.36.2.6 `uint32_t netflow_v1_record::last`

Definition at line 82 of file netflow.h.

6.36.2.7 `uint32_t netflow_v1_record::next_hop`

Definition at line 76 of file netflow.h.

6.36.2.8 `uint32_t netflow_v1_record::octets`

Definition at line 80 of file netflow.h.

6.36.2.9 `uint16_t netflow_v1_record::out_ifx`

Definition at line 78 of file netflow.h.

6.36.2.10 `uint32_t netflow_v1_record::packets`

Definition at line 79 of file netflow.h.

6.36.2.11 `uint16_t netflow_v1_record::pad1`

Definition at line 85 of file netflow.h.

6.36.2.12 `uint8_t netflow_v1_record::pad2`

Definition at line 89 of file netflow.h.

6.36.2.13 `uint16_t netflow_v1_record::pad3`

Definition at line 90 of file netflow.h.

6.36.2.14 `uint8_t netflow_v1_record::prot`

Definition at line 86 of file netflow.h.

6.36.2.15 `uint8_t netflow_v1_record::reserved[5]`

Definition at line 91 of file netflow.h.

6.36.2.16 `uint16_t netflow_v1_record::s_port`

Definition at line 83 of file netflow.h.

6.36.2.17 `uint32_t netflow_v1_record::src_addr`

Definition at line 74 of file netflow.h.

6.36.2.18 `uint8_t netflow_v1_record::tos`

Definition at line 87 of file netflow.h.

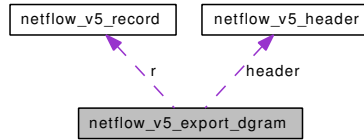
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/netflow.h](#)

6.37 netflow_v5_export_dgram Struct Reference

```
#include <netflow.h>
```

Collaboration diagram for netflow_v5_export_dgram:



Data Fields

- [netflow_v5_header header](#)
- [netflow_v5_record r](#) [NETFLOW_V5_MAX_RECORDS]

6.37.1 Detailed Description

Definition at line 126 of file netflow.h.

6.37.2 Field Documentation

6.37.2.1 struct [netflow_v5_header](#) [netflow_v5_export_dgram::header](#)

Definition at line 127 of file netflow.h.

Referenced by [export_add\(\)](#), [export_send\(\)](#), and [get_export_dgram\(\)](#).

6.37.2.2 struct [netflow_v5_record](#) [netflow_v5_export_dgram::r](#)[NETFLOW_V5_MAX_RECORDS]

Definition at line 128 of file netflow.h.

Referenced by [export_add\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/netflow.h](#)

6.38 netflow_v5_header Struct Reference

```
#include <netflow.h>
```

Data Fields

- [uint16_t version](#)
- [uint16_t count](#)
- [uint32_t sys_uptime](#)
- [uint32_t unix_secs](#)
- [uint32_t unix_nsecs](#)
- [uint32_t flow_seq](#)
- [uint8_t engine_type](#)
- [uint8_t engine_id](#)
- [uint16_t pad](#)

6.38.1 Detailed Description

Definition at line 59 of file netflow.h.

6.38.2 Field Documentation

6.38.2.1 [uint16_t netflow_v5_header::count](#)

Definition at line 62 of file netflow.h.

Referenced by [export_add\(\)](#), [export_send\(\)](#), and [get_export_dgram\(\)](#).

6.38.2.2 [uint8_t netflow_v5_header::engine_id](#)

Definition at line 68 of file netflow.h.

Referenced by [export_send\(\)](#).

6.38.2.3 [uint8_t netflow_v5_header::engine_type](#)

Definition at line 67 of file netflow.h.

Referenced by [export_send\(\)](#).

6.38.2.4 [uint32_t netflow_v5_header::flow_seq](#)

Definition at line 66 of file netflow.h.

Referenced by [export_send\(\)](#).

6.38.2.5 [uint16_t netflow_v5_header::pad](#)

Definition at line 69 of file netflow.h.

Referenced by [export_send\(\)](#).

6.38.2.6 `uint32_t netflow_v5_header::sys_uptime`

Definition at line 63 of file netflow.h.

Referenced by `export_send()`.

6.38.2.7 `uint32_t netflow_v5_header::unix_nsecs`

Definition at line 65 of file netflow.h.

Referenced by `export_send()`.

6.38.2.8 `uint32_t netflow_v5_header::unix_secs`

Definition at line 64 of file netflow.h.

Referenced by `export_send()`.

6.38.2.9 `uint16_t netflow_v5_header::version`

Definition at line 61 of file netflow.h.

Referenced by `get_export_dgram()`.

The documentation for this struct was generated from the following file:

- </usr/src/sys/netgraph/netflow/netflow.h>

6.39 netflow_v5_record Struct Reference

```
#include <netflow.h>
```

Data Fields

- [uint32_t src_addr](#)
- [uint32_t dst_addr](#)
- [uint32_t next_hop](#)
- [uint16_t i_ifx](#)
- [uint16_t o_ifx](#)
- [uint32_t packets](#)
- [uint32_t octets](#)
- [uint32_t first](#)
- [uint32_t last](#)
- [uint16_t s_port](#)
- [uint16_t d_port](#)
- [uint8_t pad1](#)
- [uint8_t flags](#)
- [uint8_t prot](#)
- [uint8_t tos](#)
- [uint16_t src_as](#)
- [uint16_t dst_as](#)
- [uint8_t src_mask](#)
- [uint8_t dst_mask](#)
- [uint16_t pad2](#)

6.39.1 Detailed Description

Definition at line 94 of file netflow.h.

6.39.2 Field Documentation

6.39.2.1 [uint16_t netflow_v5_record::d_port](#)

Definition at line 106 of file netflow.h.

Referenced by [export_add\(\)](#).

6.39.2.2 [uint32_t netflow_v5_record::dst_addr](#)

Definition at line 97 of file netflow.h.

Referenced by [export_add\(\)](#).

6.39.2.3 [uint16_t netflow_v5_record::dst_as](#)

Definition at line 112 of file netflow.h.

Referenced by [export_add\(\)](#).

6.39.2.4 `uint8_t netflow_v5_record::dst_mask`

Definition at line 114 of file netflow.h.

Referenced by `export_add()`.

6.39.2.5 `uint32_t netflow_v5_record::first`

Definition at line 103 of file netflow.h.

Referenced by `export_add()`.

6.39.2.6 `uint8_t netflow_v5_record::flags`

Definition at line 108 of file netflow.h.

Referenced by `export_add()`.

6.39.2.7 `uint16_t netflow_v5_record::i_ifx`

Definition at line 99 of file netflow.h.

Referenced by `export_add()`.

6.39.2.8 `uint32_t netflow_v5_record::last`

Definition at line 104 of file netflow.h.

Referenced by `export_add()`.

6.39.2.9 `uint32_t netflow_v5_record::next_hop`

Definition at line 98 of file netflow.h.

Referenced by `export_add()`.

6.39.2.10 `uint16_t netflow_v5_record::o_ifx`

Definition at line 100 of file netflow.h.

Referenced by `export_add()`.

6.39.2.11 `uint32_t netflow_v5_record::octets`

Definition at line 102 of file netflow.h.

Referenced by `export_add()`.

6.39.2.12 `uint32_t netflow_v5_record::packets`

Definition at line 101 of file netflow.h.

Referenced by `export_add()`.

6.39.2.13 `uint8_t netflow_v5_record::pad1`

Definition at line 107 of file netflow.h.

6.39.2.14 `uint16_t netflow_v5_record::pad2`

Definition at line 115 of file netflow.h.

6.39.2.15 `uint8_t netflow_v5_record::prot`

Definition at line 109 of file netflow.h.

Referenced by `export_add()`.

6.39.2.16 `uint16_t netflow_v5_record::s_port`

Definition at line 105 of file netflow.h.

Referenced by `export_add()`.

6.39.2.17 `uint32_t netflow_v5_record::src_addr`

Definition at line 96 of file netflow.h.

Referenced by `export_add()`.

6.39.2.18 `uint16_t netflow_v5_record::src_as`

Definition at line 111 of file netflow.h.

Referenced by `export_add()`.

6.39.2.19 `uint8_t netflow_v5_record::src_mask`

Definition at line 113 of file netflow.h.

Referenced by `export_add()`.

6.39.2.20 `uint8_t netflow_v5_record::tos`

Definition at line 110 of file netflow.h.

Referenced by `export_add()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/netflow.h`

6.40 ng_async_cfg Struct Reference

```
#include <ng_async.h>
```

Data Fields

- u_char [enabled](#)
- u_int16_t [amru](#)
- u_int16_t [smru](#)
- u_int32_t [accm](#)

6.40.1 Detailed Description

Definition at line 86 of file `ng_async.h`.

6.40.2 Field Documentation

6.40.2.1 u_int32_t [ng_async_cfg::accm](#)

Definition at line 90 of file `ng_async.h`.

Referenced by `nga_rcv_sync()`.

6.40.2.2 u_int16_t [ng_async_cfg::amru](#)

Definition at line 88 of file `ng_async.h`.

Referenced by `nga_rcv_async()`, and `nga_rcvmsg()`.

6.40.2.3 u_char [ng_async_cfg::enabled](#)

Definition at line 87 of file `ng_async.h`.

Referenced by `nga_rcv_async()`, and `nga_rcv_sync()`.

6.40.2.4 u_int16_t [ng_async_cfg::smru](#)

Definition at line 89 of file `ng_async.h`.

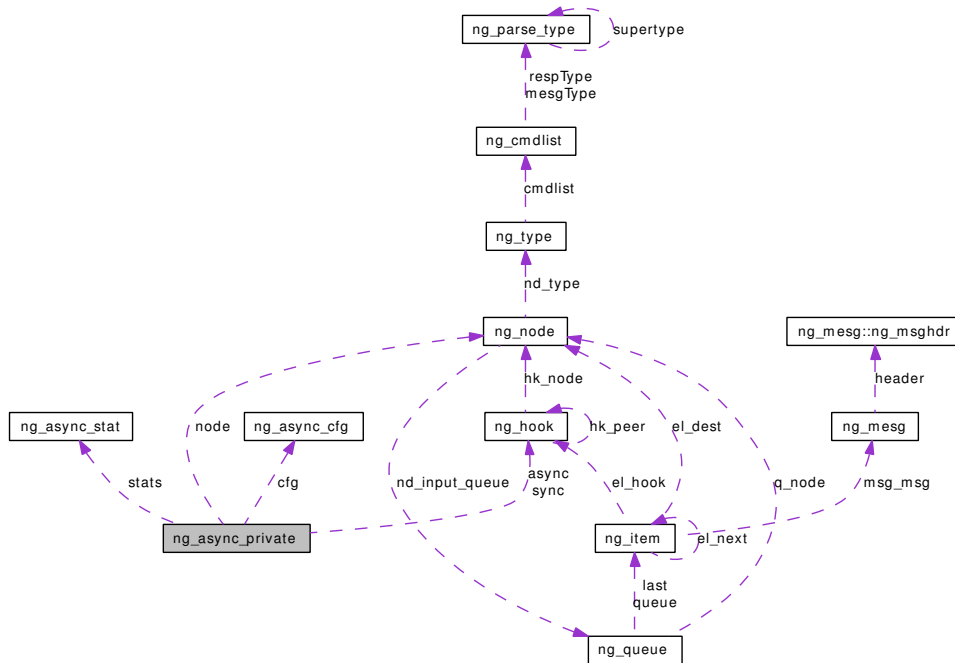
Referenced by `nga_rcv_sync()`, and `nga_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_async.h`

6.41 ng_async_private Struct Reference

Collaboration diagram for ng_async_private:



Data Fields

- `node_p` node
- `hook_p` async
- `hook_p` sync
- `u_char` amode
- `u_int16_t` fcs
- `u_char *` abuf
- `u_char *` sbuf
- `u_int` slen
- `long` lasttime
- `ng_async_cfg` cfg
- `ng_async_stat` stats

6.41.1 Detailed Description

Definition at line 76 of file `ng_async.c`.

6.41.2 Field Documentation

6.41.2.1 `u_char*` `ng_async_private::abuf`

Definition at line 82 of file `ng_async.c`.

Referenced by `nga_async_add()`, `nga_rcv_sync()`, `nga_rcvmsg()`, and `nga_shutdown()`.

6.41.2.2 `u_char ng_async_private::amode`

Definition at line 80 of file `ng_async.c`.

Referenced by `nga_rcvmsg()`.

6.41.2.3 `hook_p ng_async_private::async`

Definition at line 78 of file `ng_async.c`.

Referenced by `nga_disconnect()`, `nga_newhook()`, `nga_rcv_sync()`, and `nga_rcvdata()`.

6.41.2.4 `struct ng_async_cfg ng_async_private::cfg`

Definition at line 86 of file `ng_async.c`.

Referenced by `nga_rcv_async()`, `nga_rcv_sync()`, and `nga_rcvmsg()`.

6.41.2.5 `u_int16_t ng_async_private::fcs`

Definition at line 81 of file `ng_async.c`.

Referenced by `nga_rcv_async()`.

6.41.2.6 `long ng_async_private::lasttime`

Definition at line 85 of file `ng_async.c`.

Referenced by `nga_disconnect()`, and `nga_rcv_sync()`.

6.41.2.7 `node_p ng_async_private::node`

Definition at line 77 of file `ng_async.c`.

Referenced by `cisco_input()`, `cisco_notify()`, `cisco_shutdown()`, `ng_source_connect()`, `ng_source_intr()`, `ng_source_rcvmsg()`, `ng_source_send()`, `ng_source_set_autosrc()`, `ng_source_start()`, `ng_source_stop()`, `nglmi_disconnect()`, `nglmi_shutdown()`, `nglmi_startup()`, `ngt_input()`, `ngt_shutdown()`, `ngt_start()`, and `ngt_tioctl()`.

6.41.2.8 `u_char* ng_async_private::sbuf`

Definition at line 83 of file `ng_async.c`.

Referenced by `nga_rcv_async()`, `nga_rcvmsg()`, and `nga_shutdown()`.

6.41.2.9 `u_int ng_async_private::slen`

Definition at line 84 of file `ng_async.c`.

Referenced by `nga_rcv_async()`, and `nga_rcvmsg()`.

6.41.2.10 struct [ng_async_stat](#) [ng_async_private::stats](#)

Definition at line 87 of file [ng_async.c](#).

Referenced by [ng_source_intr\(\)](#), [ng_source_newhook\(\)](#), [ng_source_rcvmsg\(\)](#), [ng_source_send\(\)](#), [ng_source_start\(\)](#), [ng_source_stop\(\)](#), [nga_disconnect\(\)](#), [nga_rcv_async\(\)](#), [nga_rcv_sync\(\)](#), [nga_rcvmsg\(\)](#), [ngt_newhook\(\)](#), and [ngt_rcvmsg\(\)](#).

6.41.2.11 hook_p [ng_async_private::sync](#)

Definition at line 79 of file [ng_async.c](#).

Referenced by [nga_disconnect\(\)](#), [nga_newhook\(\)](#), [nga_rcv_async\(\)](#), and [nga_rcvdata\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_async.c](#)

6.42 ng_async_stat Struct Reference

```
#include <ng_async.h>
```

Data Fields

- [u_int32_t syncOctets](#)
- [u_int32_t syncFrames](#)
- [u_int32_t syncOverflows](#)
- [u_int32_t asyncOctets](#)
- [u_int32_t asyncFrames](#)
- [u_int32_t asyncRunts](#)
- [u_int32_t asyncOverflows](#)
- [u_int32_t asyncBadCheckSums](#)

6.42.1 Detailed Description

Definition at line 61 of file `ng_async.h`.

6.42.2 Field Documentation

6.42.2.1 [u_int32_t ng_async_stat::asyncBadCheckSums](#)

Definition at line 69 of file `ng_async.h`.

Referenced by `nga_rcv_async()`.

6.42.2.2 [u_int32_t ng_async_stat::asyncFrames](#)

Definition at line 66 of file `ng_async.h`.

6.42.2.3 [u_int32_t ng_async_stat::asyncOctets](#)

Definition at line 65 of file `ng_async.h`.

Referenced by `nga_rcv_async()`.

6.42.2.4 [u_int32_t ng_async_stat::asyncOverflows](#)

Definition at line 68 of file `ng_async.h`.

Referenced by `nga_rcv_async()`.

6.42.2.5 [u_int32_t ng_async_stat::asyncRunts](#)

Definition at line 67 of file `ng_async.h`.

Referenced by `nga_rcv_async()`.

6.42.2.6 `u_int32_t ng_async_stat::syncFrames`

Definition at line 63 of file `ng_async.h`.

Referenced by `nga_rcv_sync()`.

6.42.2.7 `u_int32_t ng_async_stat::syncOctets`

Definition at line 62 of file `ng_async.h`.

Referenced by `nga_rcv_sync()`.

6.42.2.8 `u_int32_t ng_async_stat::syncOverflows`

Definition at line 64 of file `ng_async.h`.

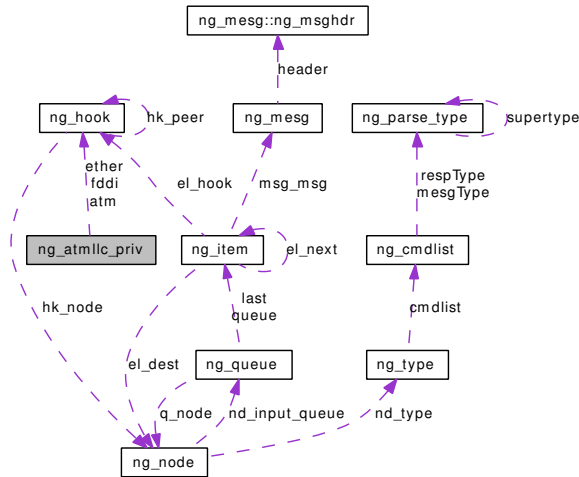
Referenced by `nga_rcv_sync()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_async.h`

6.43 ng_atmllc_priv Struct Reference

Collaboration diagram for ng_atmllc_priv:



Data Fields

- `hook_p atm`
- `hook_p ether`
- `hook_p fddi`

6.43.1 Detailed Description

Definition at line 52 of file `ng_atmllc.c`.

6.43.2 Field Documentation

6.43.2.1 `hook_p ng_atmllc_priv::atm`

Definition at line 53 of file `ng_atmllc.c`.

Referenced by `ng_atmllc_disconnect()`, `ng_atmllc_newhook()`, and `ng_atmllc_rcvdata()`.

6.43.2.2 `hook_p ng_atmllc_priv::ether`

Definition at line 54 of file `ng_atmllc.c`.

Referenced by `ng_atmllc_disconnect()`, `ng_atmllc_newhook()`, and `ng_atmllc_rcvdata()`.

6.43.2.3 `hook_p ng_atmllc_priv::fddi`

Definition at line 55 of file `ng_atmllc.c`.

Referenced by `ng_atmllc_disconnect()`, `ng_atmllc_newhook()`, and `ng_atmllc_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_atmllc.c](#)

6.44 ng_atmpif_link_status Struct Reference

```
#include <ng_atmpif.h>
```

Data Fields

- [uint32_t InSeq](#)
- [uint32_t OutSeq](#)
- [uint32_t cur_pcr](#)

6.44.1 Detailed Description

Definition at line 152 of file `ng_atmpif.h`.

6.44.2 Field Documentation

6.44.2.1 [uint32_t ng_atmpif_link_status::cur_pcr](#)

Definition at line 155 of file `ng_atmpif.h`.

6.44.2.2 [uint32_t ng_atmpif_link_status::InSeq](#)

Definition at line 153 of file `ng_atmpif.h`.

6.44.2.3 [uint32_t ng_atmpif_link_status::OutSeq](#)

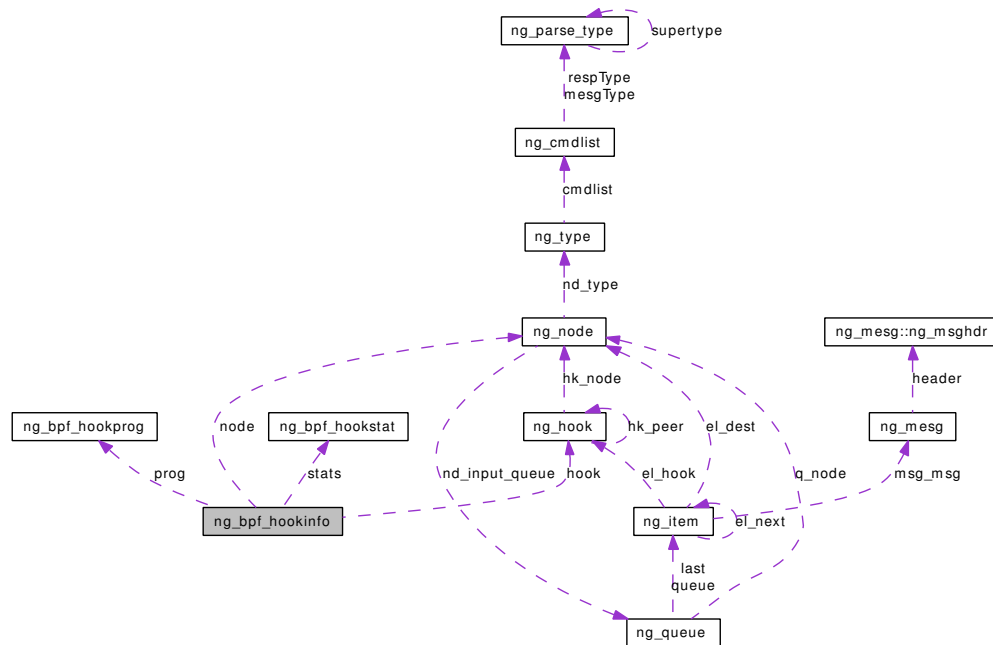
Definition at line 154 of file `ng_atmpif.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atmpif.h](#)

6.45 ng_bpf_hookinfo Struct Reference

Collaboration diagram for ng_bpf_hookinfo:



Data Fields

- `node_p` `node`
- `hook_p` `hook`
- `ng_bpf_hookprog` * `prog`
- `ng_bpf_hookstat` `stats`

6.45.1 Detailed Description

Definition at line 87 of file `ng_bpf.c`.

6.45.2 Field Documentation

6.45.2.1 `hook_p` `ng_bpf_hookinfo::hook`

Definition at line 89 of file `ng_bpf.c`.

6.45.2.2 `node_p` `ng_bpf_hookinfo::node`

Definition at line 88 of file `ng_bpf.c`.

Referenced by `ng_bpf_rcvdata()`.

6.45.2.3 struct [ng_bpf_hookprog](#)* [ng_bpf_hookinfo::prog](#)

Definition at line 90 of file [ng_bpf.c](#).

Referenced by [ng_bpf_disconnect\(\)](#), [ng_bpf_rcvdata\(\)](#), and [ng_bpf_setprog\(\)](#).

6.45.2.4 struct [ng_bpf_hookstat](#) [ng_bpf_hookinfo::stats](#)

Definition at line 94 of file [ng_bpf.c](#).

Referenced by [ng_bpf_rcvdata\(\)](#), [ng_tag_rcvdata\(\)](#), and [ngh_rcvdata\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_bpf.c](#)

6.46 ng_bpf_hookprog Struct Reference

```
#include <ng_bpf.h>
```

Data Fields

- char [thisHook](#) [NG_HOOKSIZ]
- char [ifMatch](#) [NG_HOOKSIZ]
- char [ifNotMatch](#) [NG_HOOKSIZ]
- int32_t [bpf_prog_len](#)
- bpf_insn [bpf_prog](#) []

6.46.1 Detailed Description

Definition at line 52 of file [ng_bpf.h](#).

6.46.2 Field Documentation

6.46.2.1 struct bpf_insn [ng_bpf_hookprog::bpf_prog](#) []

Definition at line 57 of file [ng_bpf.h](#).

Referenced by [ng_bpf_hookprogary_getLength\(\)](#), [ng_bpf_rcvdata\(\)](#), and [ng_bpf_setprog\(\)](#).

6.46.2.2 int32_t [ng_bpf_hookprog::bpf_prog_len](#)

Definition at line 56 of file [ng_bpf.h](#).

Referenced by [ng_bpf_hookprogary_getLength\(\)](#), [ng_bpf_rcvmsg\(\)](#), and [ng_bpf_setprog\(\)](#).

6.46.2.3 char [ng_bpf_hookprog::ifMatch](#)[NG_HOOKSIZ]

Definition at line 54 of file [ng_bpf.h](#).

Referenced by [ng_bpf_rcvdata\(\)](#).

6.46.2.4 char [ng_bpf_hookprog::ifNotMatch](#)[NG_HOOKSIZ]

Definition at line 55 of file [ng_bpf.h](#).

Referenced by [ng_bpf_rcvdata\(\)](#).

6.46.2.5 char [ng_bpf_hookprog::thisHook](#)[NG_HOOKSIZ]

Definition at line 53 of file [ng_bpf.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_bpf.h](#)

6.47 ng_bpf_hookstat Struct Reference

```
#include <ng_bpf.h>
```

Data Fields

- [u_int64_t recvFrames](#)
- [u_int64_t recvOctets](#)
- [u_int64_t recvMatchFrames](#)
- [u_int64_t recvMatchOctets](#)
- [u_int64_t xmitFrames](#)
- [u_int64_t xmitOctets](#)

6.47.1 Detailed Description

Definition at line 74 of file ng_bpf.h.

6.47.2 Field Documentation

6.47.2.1 [u_int64_t ng_bpf_hookstat::recvFrames](#)

Definition at line 75 of file ng_bpf.h.

Referenced by [ng_bpf_rcvdata\(\)](#), and [ng_tag_rcvdata\(\)](#).

6.47.2.2 [u_int64_t ng_bpf_hookstat::recvMatchFrames](#)

Definition at line 77 of file ng_bpf.h.

Referenced by [ng_bpf_rcvdata\(\)](#), and [ng_tag_rcvdata\(\)](#).

6.47.2.3 [u_int64_t ng_bpf_hookstat::recvMatchOctets](#)

Definition at line 78 of file ng_bpf.h.

Referenced by [ng_bpf_rcvdata\(\)](#), and [ng_tag_rcvdata\(\)](#).

6.47.2.4 [u_int64_t ng_bpf_hookstat::recvOctets](#)

Definition at line 76 of file ng_bpf.h.

Referenced by [ng_bpf_rcvdata\(\)](#), and [ng_tag_rcvdata\(\)](#).

6.47.2.5 [u_int64_t ng_bpf_hookstat::xmitFrames](#)

Definition at line 79 of file ng_bpf.h.

Referenced by [ng_bpf_rcvdata\(\)](#), and [ng_tag_rcvdata\(\)](#).

6.47.2.6 `u_int64_t ng_bpf_hookstat::xmitOctets`

Definition at line 80 of file `ng_bpf.h`.

Referenced by `ng_bpf_rcvdata()`, and `ng_tag_rcvdata()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bpf.h`

6.48 ng_bridge_config Struct Reference

```
#include <ng_bridge.h>
```

Data Fields

- u_char [ipfw](#) [NG_BRIDGE_MAX_LINKS]
- u_char [debugLevel](#)
- u_int32_t [loopTimeout](#)
- u_int32_t [maxStaleness](#)
- u_int32_t [minStableAge](#)

6.48.1 Detailed Description

Definition at line 58 of file `ng_bridge.h`.

6.48.2 Field Documentation

6.48.2.1 u_char [ng_bridge_config::debugLevel](#)

Definition at line 60 of file `ng_bridge.h`.

6.48.2.2 u_char [ng_bridge_config::ipfw](#)[NG_BRIDGE_MAX_LINKS]

Definition at line 59 of file `ng_bridge.h`.

6.48.2.3 u_int32_t [ng_bridge_config::loopTimeout](#)

Definition at line 61 of file `ng_bridge.h`.

6.48.2.4 u_int32_t [ng_bridge_config::maxStaleness](#)

Definition at line 62 of file `ng_bridge.h`.

6.48.2.5 u_int32_t [ng_bridge_config::minStableAge](#)

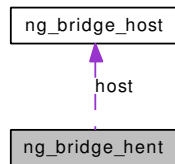
Definition at line 63 of file `ng_bridge.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bridge.h`

6.49 ng_bridge_hent Struct Reference

Collaboration diagram for ng_bridge_hent:



Data Fields

- [ng_bridge_host](#) host

6.49.1 Detailed Description

Definition at line 110 of file ng_bridge.c.

6.49.2 Field Documentation

6.49.2.1 struct [ng_bridge_host](#) [ng_bridge_hent::host](#)

Definition at line 111 of file ng_bridge.c.

Referenced by [ng_bridge_get\(\)](#), [ng_bridge_put\(\)](#), and [ng_bridge_rcvmsg\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_bridge.c](#)

6.50 ng_bridge_host Struct Reference

```
#include <ng_bridge.h>
```

Data Fields

- u_char [addr](#) [6]
- u_int16_t [linkNum](#)
- u_int16_t [age](#)
- u_int16_t [staleness](#)

6.50.1 Detailed Description

Definition at line 114 of file `ng_bridge.h`.

6.50.2 Field Documentation

6.50.2.1 u_char [ng_bridge_host::addr](#)[6]

Definition at line 115 of file `ng_bridge.h`.

Referenced by `ng_bridge_get()`, and `ng_bridge_put()`.

6.50.2.2 u_int16_t [ng_bridge_host::age](#)

Definition at line 117 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.50.2.3 u_int16_t [ng_bridge_host::linkNum](#)

Definition at line 116 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.50.2.4 u_int16_t [ng_bridge_host::staleness](#)

Definition at line 118 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

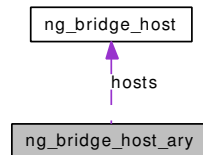
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bridge.h`

6.51 ng_bridge_host_ary Struct Reference

```
#include <ng_bridge.h>
```

Collaboration diagram for ng_bridge_host_ary:



Data Fields

- `u_int32_t numHosts`
- `ng_bridge_host hosts []`

6.51.1 Detailed Description

Definition at line 131 of file `ng_bridge.h`.

6.51.2 Field Documentation

6.51.2.1 struct `ng_bridge_host ng_bridge_host_ary::hosts[]`

Definition at line 133 of file `ng_bridge.h`.

6.51.2.2 `u_int32_t ng_bridge_host_ary::numHosts`

Definition at line 132 of file `ng_bridge.h`.

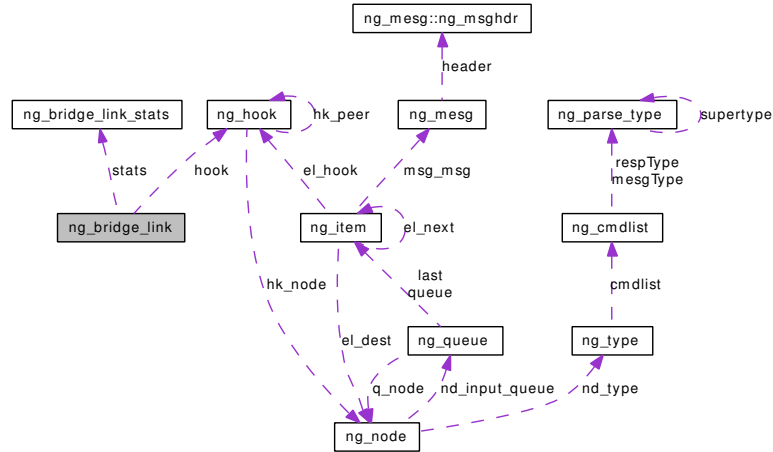
Referenced by `ng_bridge_getTableLength()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bridge.h`

6.52 ng_bridge_link Struct Reference

Collaboration diagram for ng_bridge_link:



Data Fields

- [hook_p](#) hook
- [u_int16_t](#) loopCount
- [ng_bridge_link_stats](#) stats

6.52.1 Detailed Description

Definition at line 89 of file ng_bridge.c.

6.52.2 Field Documentation

6.52.2.1 [hook_p](#) ng_bridge_link::hook

Definition at line 90 of file ng_bridge.c.

Referenced by ng_bridge_rcvdata().

6.52.2.2 [u_int16_t](#) ng_bridge_link::loopCount

Definition at line 91 of file ng_bridge.c.

Referenced by ng_bridge_rcvdata().

6.52.2.3 [struct ng_bridge_link_stats](#) ng_bridge_link::stats

Definition at line 92 of file ng_bridge.c.

Referenced by ng_bridge_rcvdata().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_bridge.c](#)

6.53 ng_bridge_link_stats Struct Reference

```
#include <ng_bridge.h>
```

Data Fields

- `u_int64_t` [recvOctets](#)
- `u_int64_t` [recvPackets](#)
- `u_int64_t` [recvMulticasts](#)
- `u_int64_t` [recvBroadcasts](#)
- `u_int64_t` [recvUnknown](#)
- `u_int64_t` [recvRunts](#)
- `u_int64_t` [recvInvalid](#)
- `u_int64_t` [xmitOctets](#)
- `u_int64_t` [xmitPackets](#)
- `u_int64_t` [xmitMulticasts](#)
- `u_int64_t` [xmitBroadcasts](#)
- `u_int64_t` [loopDrops](#)
- `u_int64_t` [loopDetects](#)
- `u_int64_t` [memoryFailures](#)

6.53.1 Detailed Description

Definition at line 77 of file `ng_bridge.h`.

6.53.2 Field Documentation

6.53.2.1 `u_int64_t` [ng_bridge_link_stats::loopDetects](#)

Definition at line 90 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.53.2.2 `u_int64_t` [ng_bridge_link_stats::loopDrops](#)

Definition at line 89 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.53.2.3 `u_int64_t` [ng_bridge_link_stats::memoryFailures](#)

Definition at line 91 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.53.2.4 `u_int64_t` [ng_bridge_link_stats::recvBroadcasts](#)

Definition at line 81 of file `ng_bridge.h`.

Referenced by `ng_bridge_rcvdata()`.

6.53.2.5 `u_int64_t ng_bridge_link_stats::recvInvalid`

Definition at line 84 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.6 `u_int64_t ng_bridge_link_stats::recvMulticasts`

Definition at line 80 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.7 `u_int64_t ng_bridge_link_stats::recvOctets`

Definition at line 78 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.8 `u_int64_t ng_bridge_link_stats::recvPackets`

Definition at line 79 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.9 `u_int64_t ng_bridge_link_stats::recvRunts`

Definition at line 83 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.10 `u_int64_t ng_bridge_link_stats::recvUnknown`

Definition at line 82 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.11 `u_int64_t ng_bridge_link_stats::xmitBroadcasts`

Definition at line 88 of file ng_bridge.h.

6.53.2.12 `u_int64_t ng_bridge_link_stats::xmitMulticasts`

Definition at line 87 of file ng_bridge.h.

6.53.2.13 `u_int64_t ng_bridge_link_stats::xmitOctets`

Definition at line 85 of file ng_bridge.h.

Referenced by ng_bridge_rcvdata().

6.53.2.14 `u_int64_t ng_bridge_link_stats::xmitPackets`

Definition at line 86 of file `ng_bridge.h`.

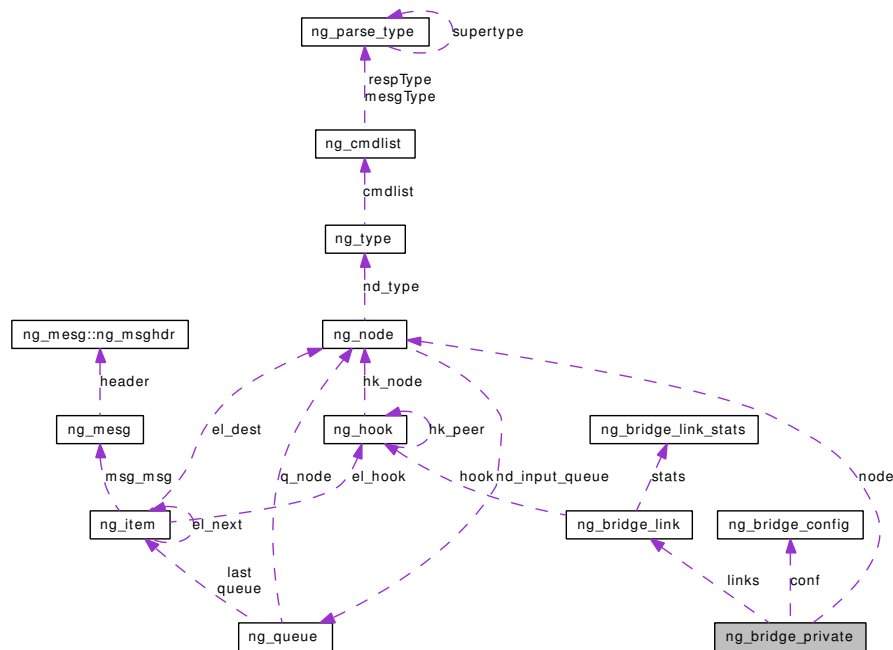
Referenced by `ng_bridge_rcvdata()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bridge.h`

6.54 ng_bridge_private Struct Reference

Collaboration diagram for ng_bridge_private:



Data Fields

- `ng_bridge_bucket` * `tab`
- `ng_bridge_link` * `links` [`NG_BRIDGE_MAX_LINKS`]
- `ng_bridge_config` `conf`
- `node_p` `node`
- `u_int` `numHosts`
- `u_int` `numBuckets`
- `u_int` `hashMask`
- `int` `numLinks`
- `callout` `timer`

6.54.1 Detailed Description

Definition at line 96 of file `ng_bridge.c`.

6.54.2 Field Documentation

6.54.2.1 struct `ng_bridge_config` `ng_bridge_private::conf`

Definition at line 99 of file `ng_bridge.c`.

6.54.2.2 `u_int ng_bridge_private::hashMask`

Definition at line 103 of file `ng_bridge.c`.

6.54.2.3 `struct ng_bridge_link* ng_bridge_private::links[NG_BRIDGE_MAX_LINKS]`

Definition at line 98 of file `ng_bridge.c`.

6.54.2.4 `node_p ng_bridge_private::node`

Definition at line 100 of file `ng_bridge.c`.

6.54.2.5 `u_int ng_bridge_private::numBuckets`

Definition at line 102 of file `ng_bridge.c`.

6.54.2.6 `u_int ng_bridge_private::numHosts`

Definition at line 101 of file `ng_bridge.c`.

6.54.2.7 `int ng_bridge_private::numLinks`

Definition at line 104 of file `ng_bridge.c`.

6.54.2.8 `struct ng_bridge_bucket* ng_bridge_private::tab`

Definition at line 97 of file `ng_bridge.c`.

6.54.2.9 `struct callout ng_bridge_private::timer`

Definition at line 105 of file `ng_bridge.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_bridge.c`

6.55 ng_bt3c_firmware_block_ep Struct Reference

```
#include <ng_bt3c.h>
```

Data Fields

- [u_int32_t block_address](#)
- [u_int16_t block_size](#)
- [u_int16_t block_alignment](#)

6.55.1 Detailed Description

Definition at line 104 of file `ng_bt3c.h`.

6.55.2 Field Documentation

6.55.2.1 [u_int32_t ng_bt3c_firmware_block_ep::block_address](#)

Definition at line 105 of file `ng_bt3c.h`.

Referenced by `bt3c_download_firmware()`.

6.55.2.2 [u_int16_t ng_bt3c_firmware_block_ep::block_alignment](#)

Definition at line 107 of file `ng_bt3c.h`.

Referenced by `bt3c_download_firmware()`.

6.55.2.3 [u_int16_t ng_bt3c_firmware_block_ep::block_size](#)

Definition at line 106 of file `ng_bt3c.h`.

Referenced by `bt3c_download_firmware()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h](#)

6.56 ng_bt3c_node_qlen_ep Struct Reference

```
#include <ng_bt3c.h>
```

Data Fields

- [int32_t queue](#)
- [int32_t qlen](#)

6.56.1 Detailed Description

Definition at line 82 of file `ng_bt3c.h`.

6.56.2 Field Documentation

6.56.2.1 [int32_t ng_bt3c_node_qlen_ep::qlen](#)

Definition at line 87 of file `ng_bt3c.h`.

6.56.2.2 [int32_t ng_bt3c_node_qlen_ep::queue](#)

Definition at line 83 of file `ng_bt3c.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h](#)

6.57 ng_bt3c_node_stat_ep Struct Reference

```
#include <ng_bt3c.h>
```

Data Fields

- [u_int32_t pckts_recv](#)
- [u_int32_t bytes_recv](#)
- [u_int32_t pckts_sent](#)
- [u_int32_t bytes_sent](#)
- [u_int32_t oerrors](#)
- [u_int32_t ierrors](#)

6.57.1 Detailed Description

Definition at line 91 of file [ng_bt3c.h](#).

6.57.2 Field Documentation

6.57.2.1 [u_int32_t ng_bt3c_node_stat_ep::bytes_recv](#)

Definition at line 93 of file [ng_bt3c.h](#).

6.57.2.2 [u_int32_t ng_bt3c_node_stat_ep::bytes_sent](#)

Definition at line 95 of file [ng_bt3c.h](#).

6.57.2.3 [u_int32_t ng_bt3c_node_stat_ep::ierrors](#)

Definition at line 97 of file [ng_bt3c.h](#).

6.57.2.4 [u_int32_t ng_bt3c_node_stat_ep::oerrors](#)

Definition at line 96 of file [ng_bt3c.h](#).

6.57.2.5 [u_int32_t ng_bt3c_node_stat_ep::pckts_recv](#)

Definition at line 92 of file [ng_bt3c.h](#).

6.57.2.6 [u_int32_t ng_bt3c_node_stat_ep::pckts_sent](#)

Definition at line 94 of file [ng_bt3c.h](#).

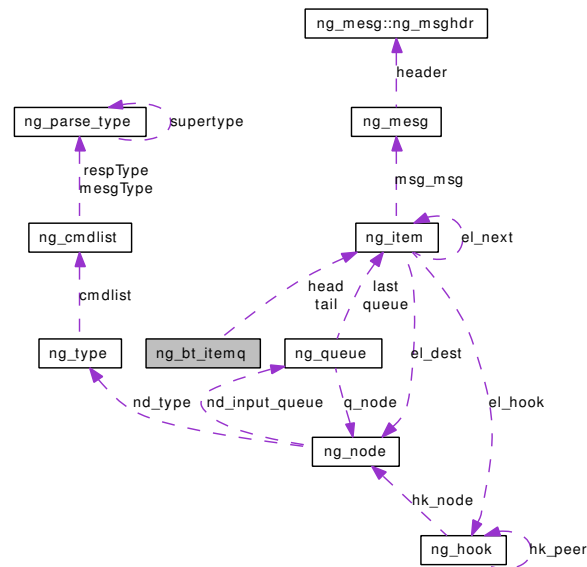
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h](#)

6.58 ng_bt_itemq Struct Reference

```
#include <ng_bluetooth.h>
```

Collaboration diagram for ng_bt_itemq:



Data Fields

- `ng_item * head`
- `ng_item * tail`
- `u_int32_t len`
- `u_int32_t maxlen`
- `u_int32_t drops`

6.58.1 Detailed Description

Definition at line 149 of file `ng_bluetooth.h`.

6.58.2 Field Documentation

6.58.2.1 `u_int32_t ng_bt_itemq::drops`

Definition at line 154 of file `ng_bluetooth.h`.

6.58.2.2 `struct ng_item* ng_bt_itemq::head`

Definition at line 150 of file `ng_bluetooth.h`.

6.58.2.3 `u_int32_t ng_bt_itemq::len`

Definition at line 152 of file `ng_bluetooth.h`.

6.58.2.4 `u_int32_t ng_bt_itemq::maxlen`

Definition at line 153 of file `ng_bluetooth.h`.

6.58.2.5 `struct ng_item* ng_bt_itemq::tail`

Definition at line 151 of file `ng_bluetooth.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h`

6.59 ng_bt_mbufq Struct Reference

```
#include <ng_bluetooth.h>
```

Data Fields

- mbuf * [head](#)
- mbuf * [tail](#)
- u_int32_t [len](#)
- u_int32_t [maxlen](#)
- u_int32_t [drops](#)

6.59.1 Detailed Description

Definition at line 62 of file `ng_bluetooth.h`.

6.59.2 Field Documentation

6.59.2.1 u_int32_t [ng_bt_mbufq::drops](#)

Definition at line 67 of file `ng_bluetooth.h`.

6.59.2.2 struct mbuf* [ng_bt_mbufq::head](#)

Definition at line 63 of file `ng_bluetooth.h`.

6.59.2.3 u_int32_t [ng_bt_mbufq::len](#)

Definition at line 65 of file `ng_bluetooth.h`.

6.59.2.4 u_int32_t [ng_bt_mbufq::maxlen](#)

Definition at line 66 of file `ng_bluetooth.h`.

Referenced by `ng_h4_rcvmsg()`, and `ng_ubt_rcvmsg()`.

6.59.2.5 struct mbuf* [ng_bt_mbufq::tail](#)

Definition at line 64 of file `ng_bluetooth.h`.

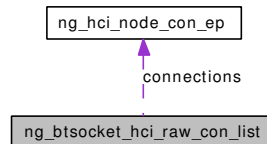
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h`

6.60 ng_btsocket_hci_raw_con_list Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_hci_raw_con_list:



Data Fields

- `u_int32_t num_connections`
- `ng_hci_node_con_ep * connections`

6.60.1 Detailed Description

Definition at line 151 of file `ng_btsocket.h`.

6.60.2 Field Documentation

6.60.2.1 `ng_hci_node_con_ep* ng_btsocket_hci_raw_con_list::connections`

Definition at line 153 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

6.60.2.2 `u_int32_t ng_btsocket_hci_raw_con_list::num_connections`

Definition at line 152 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.61 ng_btsocket_hci_raw_filter Struct Reference

```
#include <ng_btsocket.h>
```

Public Member Functions

- bitstr_t [bit_decl](#) (packet_mask, 32)
- bitstr_t [bit_decl](#) (event_mask,(NG_HCI_EVENT_MASK_SIZE *8))

6.61.1 Detailed Description

Definition at line 69 of file ng_btsocket.h.

6.61.2 Member Function Documentation

6.61.2.1 bitstr_t ng_btsocket_hci_raw_filter::bit_decl (event_mask, (NG_HCI_EVENT_MASK_SIZE *8))

6.61.2.2 bitstr_t ng_btsocket_hci_raw_filter::bit_decl (packet_mask, 32)

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.62 `ng_btsocket_hci_raw_node_bdaddr` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- `bdaddr_t bdaddr`

6.62.1 Detailed Description

Definition at line 110 of file `ng_btsocket.h`.

6.62.2 Field Documentation

6.62.2.1 `bdaddr_t ng_btsocket_hci_raw_node_bdaddr::bdaddr`

Definition at line 111 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

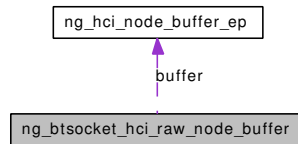
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.63 ng_btsocket_hci_raw_node_buffer Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_hci_raw_node_buffer:



Data Fields

- [ng_hci_node_buffer_ep](#) buffer

6.63.1 Detailed Description

Definition at line 102 of file ng_btsocket.h.

6.63.2 Field Documentation

6.63.2.1 [ng_hci_node_buffer_ep](#) [ng_btsocket_hci_raw_node_buffer::buffer](#)

Definition at line 103 of file ng_btsocket.h.

Referenced by [ng_btsocket_hci_raw_control\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.64 `ng_btsocket_hci_raw_node_debug` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_hci_node_debug_ep](#) debug

6.64.1 Detailed Description

Definition at line 91 of file `ng_btsocket.h`.

6.64.2 Field Documentation

6.64.2.1 [ng_hci_node_debug_ep](#) `ng_btsocket_hci_raw_node_debug::debug`

Definition at line 92 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.65 ng_btsocket_hci_raw_node_features Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- `u_int8_t features` [NG_HCI_FEATURES_SIZE]

6.65.1 Detailed Description

Definition at line 118 of file `ng_btsocket.h`.

6.65.2 Field Documentation

6.65.2.1 `u_int8_t ng_btsocket_hci_raw_node_features::features`[NG_HCI_FEATURES_SIZE]

Definition at line 119 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.66 `ng_btsocket_hci_raw_node_link_policy_mask` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_hci_node_link_policy_mask_ep policy_mask](#)

6.66.1 Detailed Description

Definition at line 160 of file `ng_btsocket.h`.

6.66.2 Field Documentation

6.66.2.1 [ng_hci_node_link_policy_mask_ep ng_btsocket_hci_raw_node_link_policy_mask::policy_mask](#)

Definition at line 161 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

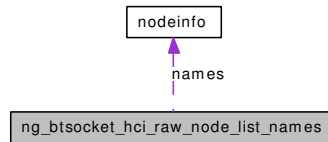
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.67 ng_btsocket_hci_raw_node_list_names Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_hci_raw_node_list_names:



Data Fields

- `u_int32_t num_names`
- `nodeinfo * names`

6.67.1 Detailed Description

Definition at line 193 of file `ng_btsocket.h`.

6.67.2 Field Documentation

6.67.2.1 `struct nodeinfo* ng_btsocket_hci_raw_node_list_names::names`

Definition at line 195 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

6.67.2.2 `u_int32_t ng_btsocket_hci_raw_node_list_names::num_names`

Definition at line 194 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

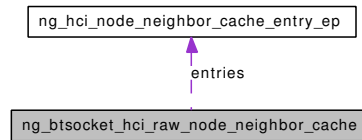
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.68 ng_btsocket_hci_raw_node_neighbor_cache Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_hci_raw_node_neighbor_cache:



Data Fields

- [u_int32_t num_entries](#)
- [ng_hci_node_neighbor_cache_entry_ep * entries](#)

6.68.1 Detailed Description

Definition at line 142 of file [ng_btsocket.h](#).

6.68.2 Field Documentation

6.68.2.1 [ng_hci_node_neighbor_cache_entry_ep * ng_btsocket_hci_raw_node_neighbor_cache::entries](#)

Definition at line 144 of file [ng_btsocket.h](#).

Referenced by [ng_btsocket_hci_raw_control\(\)](#).

6.68.2.2 [u_int32_t ng_btsocket_hci_raw_node_neighbor_cache::num_entries](#)

Definition at line 143 of file [ng_btsocket.h](#).

Referenced by [ng_btsocket_hci_raw_control\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.69 ng_btsocket_hci_raw_node_packet_mask Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_hci_node_packet_mask_ep packet_mask](#)

6.69.1 Detailed Description

Definition at line 171 of file `ng_btsocket.h`.

6.69.2 Field Documentation

6.69.2.1 [ng_hci_node_packet_mask_ep ng_btsocket_hci_raw_node_packet_mask::packet_mask](#)

Definition at line 172 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.70 `ng_btsocket_hci_raw_node_role_switch` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_hci_node_role_switch_ep role_switch](#)

6.70.1 Detailed Description

Definition at line 182 of file `ng_btsocket.h`.

6.70.2 Field Documentation

6.70.2.1 [ng_hci_node_role_switch_ep ng_btsocket_hci_raw_node_role_switch::role_switch](#)

Definition at line 183 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

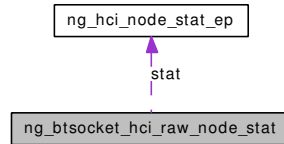
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.71 ng_btsocket_hci_raw_node_stat Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_hci_raw_node_stat:



Data Fields

- [ng_hci_node_stat_ep stat](#)

6.71.1 Detailed Description

Definition at line 126 of file ng_btsocket.h.

6.71.2 Field Documentation

6.71.2.1 [ng_hci_node_stat_ep ng_btsocket_hci_raw_node_stat::stat](#)

Definition at line 127 of file ng_btsocket.h.

Referenced by [ng_btsocket_hci_raw_control\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.72 `ng_btsocket_hci_raw_node_state` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_hci_node_state_ep state](#)

6.72.1 Detailed Description

Definition at line 79 of file `ng_btsocket.h`.

6.72.2 Field Documentation

6.72.2.1 [ng_hci_node_state_ep ng_btsocket_hci_raw_node_state::state](#)

Definition at line 80 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

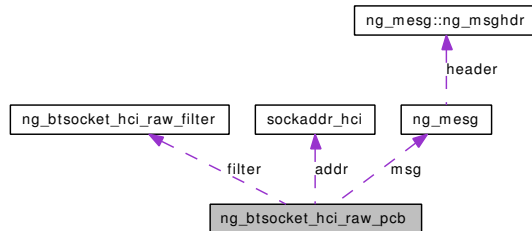
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.73 ng_btsocket_hci_raw_pcb Struct Reference

```
#include <ng_btsocket_hci_raw.h>
```

Collaboration diagram for ng_btsocket_hci_raw_pcb:



Public Member Functions

- [LIST_ENTRY \(ng_btsocket_hci_raw_pcb\) next](#)

Data Fields

- [socket * so](#)
- [u_int32_t flags](#)
- [sockaddr_hci addr](#)
- [ng_btsocket_hci_raw_filter filter](#)
- [u_int32_t token](#)
- [ng_mesg * msg](#)
- [mtx pcb_mtx](#)

6.73.1 Detailed Description

Definition at line 44 of file ng_btsocket_hci_raw.h.

6.73.2 Member Function Documentation

6.73.2.1 ng_btsocket_hci_raw_pcb::LIST_ENTRY (ng_btsocket_hci_raw_pcb)

6.73.3 Field Documentation

6.73.3.1 struct sockaddr_hci ng_btsocket_hci_raw_pcb::addr

Definition at line 49 of file ng_btsocket_hci_raw.h.

Referenced by [ng_btsocket_hci_raw_bind\(\)](#), [ng_btsocket_hci_raw_connect\(\)](#), [ng_btsocket_hci_raw_control\(\)](#), [ng_btsocket_hci_raw_data_input\(\)](#), [ng_btsocket_hci_raw_send\(\)](#), and [ng_btsocket_hci_raw_sockaddr\(\)](#).

6.73.3.2 `struct ng_btsocket_hci_raw_filter ng_btsocket_hci_raw_pcb::filter`

Definition at line 50 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_ctloutput()`, and `ng_btsocket_hci_raw_filter()`.

6.73.3.3 `u_int32_t ng_btsocket_hci_raw_pcb::flags`

Definition at line 46 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_ctloutput()`, `ng_btsocket_hci_raw_filter()`, and `ng_btsocket_hci_raw_savctl()`.

6.73.3.4 `struct ng_msg* ng_btsocket_hci_raw_pcb::msg`

Definition at line 52 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_msg_input()`, and `ng_btsocket_hci_raw_send_sync_ngmsg()`.

6.73.3.5 `struct mtx ng_btsocket_hci_raw_pcb::pcb_mtx`

Definition at line 54 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_bind()`, `ng_btsocket_hci_raw_connect()`, `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_ctloutput()`, `ng_btsocket_hci_raw_data_input()`, `ng_btsocket_hci_raw_detach()`, `ng_btsocket_hci_raw_disconnect()`, `ng_btsocket_hci_raw_filter()`, `ng_btsocket_hci_raw_msg_input()`, `ng_btsocket_hci_raw_savctl()`, `ng_btsocket_hci_raw_send()`, `ng_btsocket_hci_raw_send_sync_ngmsg()`, and `ng_btsocket_hci_raw_sockaddr()`.

6.73.3.6 `struct socket* ng_btsocket_hci_raw_pcb::so`

Definition at line 45 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_data_input()`, and `ng_btsocket_hci_raw_savctl()`.

6.73.3.7 `u_int32_t ng_btsocket_hci_raw_pcb::token`

Definition at line 51 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_msg_input()`, and `ng_btsocket_hci_raw_send_sync_ngmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_hci_raw.h`

6.74 ng_btsocket_hci_raw_sec_filter Struct Reference

Public Member Functions

- bitstr_t [bit_decl](#) (events, 0xff)
- bitstr_t [bit_decl](#) (commands[0x3f], 0x3ff)

6.74.1 Detailed Description

Definition at line 91 of file `ng_btsocket_hci_raw.c`.

6.74.2 Member Function Documentation

6.74.2.1 bitstr_t `ng_btsocket_hci_raw_sec_filter::bit_decl` (commands[0x3f], 0x3ff)

6.74.2.2 bitstr_t `ng_btsocket_hci_raw_sec_filter::bit_decl` (events, 0xff)

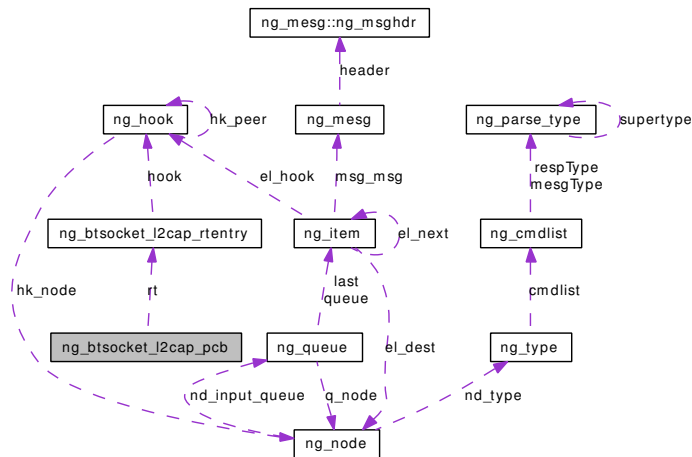
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_hci_raw.c`

6.75 ng_btsocket_l2cap_pcb Struct Reference

```
#include <ng_btsocket_l2cap.h>
```

Collaboration diagram for ng_btsocket_l2cap_pcb:



Public Member Functions

- [LIST_ENTRY \(ng_btsocket_l2cap_pcb\) next](#)

Data Fields

- socket * [so](#)
- [bdaddr_t src](#)
- [bdaddr_t dst](#)
- [u_int16_t psm](#)
- [u_int16_t cid](#)
- [u_int16_t flags](#)
- [u_int8_t state](#)
- [u_int8_t cfg_state](#)
- [u_int16_t imtu](#)
- [ng_l2cap_flow_t iflow](#)
- [u_int16_t omtu](#)
- [ng_l2cap_flow_t oflow](#)
- [u_int16_t flush_timo](#)
- [u_int16_t link_timo](#)
- [callout_handle timo](#)
- [u_int32_t token](#)
- [ng_btsocket_l2cap_rtenry_p rt](#)
- [mtx pcb_mtx](#)

6.75.1 Detailed Description

Definition at line 127 of file ng_btsocket_l2cap.h.

6.75.2 Member Function Documentation

6.75.2.1 `ng_btsocket_l2cap_pcb::LIST_ENTRY` (`ng_btsocket_l2cap_pcb`)

6.75.3 Field Documentation

6.75.3.1 `u_int8_t ng_btsocket_l2cap_pcb::cfg_state`

Definition at line 147 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, and `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`.

6.75.3.2 `u_int16_t ng_btsocket_l2cap_pcb::cid`

Definition at line 134 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_pcb_by_cid()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_process_timeout()`, `ng_btsocket_l2cap_send2()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, and `ng_btsocket_l2cap_send_l2ca_discon_req()`.

6.75.3.3 `bdaddr_t ng_btsocket_l2cap_pcb::dst`

Definition at line 131 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_peeraddr()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_session_accept()`, and `ng_btsocket_rfcomm_session_by_addr()`.

6.75.3.4 `u_int16_t ng_btsocket_l2cap_pcb::flags`

Definition at line 136 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_detach()`, `ng_btsocket_l2cap_disconnect()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_timeout()`, `ng_btsocket_l2cap_rtclean()`, `ng_btsocket_l2cap_send()`, `ng_btsocket_l2cap_timeout()`, and `ng_btsocket_l2cap_untimeout()`.

6.75.3.5 `u_int16_t ng_btsocket_l2cap_pcb::flush_timo`

Definition at line 162 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_ctloutput()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, and `ng_btsocket_l2cap_`

send_l2ca_cfg_req().

6.75.3.6 ng_l2cap_flow_t ng_btsocket_l2cap_pcb::iflow

Definition at line 157 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_process_l2ca_cfg_ind(), and ng_btsocket_l2cap_send_l2ca_cfg_rsp().

6.75.3.7 u_int16_t ng_btsocket_l2cap_pcb::imtu

Definition at line 156 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_rfcomm_session_accept(), and ng_btsocket_rfcomm_session_connect().

6.75.3.8 u_int16_t ng_btsocket_l2cap_pcb::link_timo

Definition at line 163 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_send_l2ca_cfg_req().

6.75.3.9 ng_l2cap_flow_t ng_btsocket_l2cap_pcb::oflow

Definition at line 160 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_ind(), and ng_btsocket_l2cap_send_l2ca_cfg_req().

6.75.3.10 u_int16_t ng_btsocket_l2cap_pcb::omtu

Definition at line 159 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_send(), ng_btsocket_l2cap_send_l2ca_cfg_rsp(), ng_btsocket_rfcomm_session_accept(), and ng_btsocket_rfcomm_session_connect().

6.75.3.11 struct mtx ng_btsocket_l2cap_pcb::pcb_mtx

Definition at line 170 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_connect(), ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_detach(), ng_btsocket_l2cap_disconnect(), ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_discon_ind(), ng_btsocket_l2cap_process_l2ca_discon_rsp(), ng_btsocket_l2cap_process_l2ca_write_rsp(), ng_btsocket_l2cap_process_timeout(), ng_btsocket_l2cap_rtclean(), ng_btsocket_l2cap_send(), ng_btsocket_l2cap_send2(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_l2cap_send_l2ca_cfg_rsp(), ng_btsocket_l2cap_send_l2ca_con_req(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_btsocket_l2cap_timeout(), and ng_btsocket_l2cap_untimeout().

6.75.3.12 [u_int16_t ng_btsocket_l2cap_pcb::psm](#)

Definition at line 133 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_bind()`, `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_listen()`, `ng_btsocket_l2cap_pcb_by_addr()`, `ng_btsocket_l2cap_peeraddr()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, and `ng_btsocket_l2cap_sockaddr()`.

6.75.3.13 [ng_btsocket_l2cap_rtbody_p ng_btsocket_l2cap_pcb::rt](#)

Definition at line 168 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_rtbodyclean()`, `ng_btsocket_l2cap_send()`, `ng_btsocket_l2cap_send2()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, and `ng_btsocket_l2cap_send_l2ca_discon_req()`.

6.75.3.14 [struct socket* ng_btsocket_l2cap_pcb::so](#)

Definition at line 128 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_pcb_by_addr()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_process_timeout()`, `ng_btsocket_l2cap_send()`, and `ng_btsocket_l2cap_send2()`.

6.75.3.15 [bdaddr_t ng_btsocket_l2cap_pcb::src](#)

Definition at line 130 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_bind()`, `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_pcb_by_addr()`, `ng_btsocket_l2cap_pcb_by_cid()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_sockaddr()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_session_accept()`, and `ng_btsocket_rfcomm_session_by_addr()`.

6.75.3.16 [u_int8_t ng_btsocket_l2cap_pcb::state](#)

Definition at line 140 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_ctloutput()`, `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_disconnect()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_btsocket_l2cap_send_l2ca_write_req()`, `ng_btsocket_l2cap_sockaddr()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_session_accept()`, `ng_btsocket_rfcomm_session_by_addr()`, and `ng_btsocket_rfcomm_session_timeout()`.

`l2ca_write_rsp()`, `ng_btsocket_l2cap_process_timeout()`, `ng_btsocket_l2cap_send()`, and `ng_btsocket_l2cap_send2()`.

6.75.3.17 struct `callout_handle ng_btsocket_l2cap_pcb::timo`

Definition at line 165 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_timeout()`, and `ng_btsocket_l2cap_untimeout()`.

6.75.3.18 `u_int32_t ng_btsocket_l2cap_pcb::token`

Definition at line 167 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_pcb_by_token()`, `ng_btsocket_l2cap_send2()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, and `ng_btsocket_l2cap_send_l2ca_con_req()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h`

6.76 `ng_btsocket_l2cap_raw_auto_discon_timo` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_l2cap_node_auto_discon_ep](#) timeout

6.76.1 Detailed Description

Definition at line 285 of file `ng_btsocket.h`.

6.76.2 Field Documentation

6.76.2.1 [ng_l2cap_node_auto_discon_ep](#) `ng_btsocket_l2cap_raw_auto_discon_timo::timeout`

Definition at line 287 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

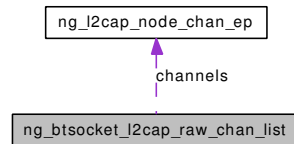
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.77 ng_btsocket_l2cap_raw_chan_list Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_l2cap_raw_chan_list:



Data Fields

- [u_int32_t num_channels](#)
- [ng_l2cap_node_chan_ep * channels](#)

6.77.1 Detailed Description

Definition at line 276 of file ng_btsocket.h.

6.77.2 Field Documentation

6.77.2.1 [ng_l2cap_node_chan_ep* ng_btsocket_l2cap_raw_chan_list::channels](#)

Definition at line 278 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

6.77.2.2 [u_int32_t ng_btsocket_l2cap_raw_chan_list::num_channels](#)

Definition at line 277 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

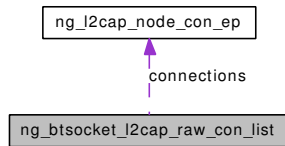
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.78 ng_btsocket_l2cap_raw_con_list Struct Reference

```
#include <ng_btsocket.h>
```

Collaboration diagram for ng_btsocket_l2cap_raw_con_list:



Data Fields

- `u_int32_t num_connections`
- `ng_l2cap_node_con_ep * connections`

6.78.1 Detailed Description

Definition at line 267 of file `ng_btsocket.h`.

6.78.2 Field Documentation

6.78.2.1 `ng_l2cap_node_con_ep* ng_btsocket_l2cap_raw_con_list::connections`

Definition at line 269 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.78.2.2 `u_int32_t ng_btsocket_l2cap_raw_con_list::num_connections`

Definition at line 268 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.79 `ng_btsocket_l2cap_raw_get_info` Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- `u_int32_t` [result](#)
- `u_int32_t` [info_type](#)
- `u_int32_t` [info_size](#)
- `u_int8_t *` [info_data](#)

6.79.1 Detailed Description

Definition at line 237 of file `ng_btsocket.h`.

6.79.2 Field Documentation

6.79.2.1 `u_int8_t *` [ng_btsocket_l2cap_raw_get_info::info_data](#)

Definition at line 241 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.79.2.2 `u_int32_t` [ng_btsocket_l2cap_raw_get_info::info_size](#)

Definition at line 240 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.79.2.3 `u_int32_t` [ng_btsocket_l2cap_raw_get_info::info_type](#)

Definition at line 239 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.79.2.4 `u_int32_t` [ng_btsocket_l2cap_raw_get_info::result](#)

Definition at line 238 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.80 ng_btsocket_l2cap_raw_node_debug Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_l2cap_node_debug_ep debug](#)

6.80.1 Detailed Description

Definition at line 256 of file `ng_btsocket.h`.

6.80.2 Field Documentation

6.80.2.1 [ng_l2cap_node_debug_ep ng_btsocket_l2cap_raw_node_debug::debug](#)

Definition at line 257 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.81 ng_btsocket_l2cap_raw_node_flags Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [ng_l2cap_node_flags_ep flags](#)

6.81.1 Detailed Description

Definition at line 248 of file ng_btsocket.h.

6.81.2 Field Documentation

6.81.2.1 [ng_l2cap_node_flags_ep ng_btsocket_l2cap_raw_node_flags::flags](#)

Definition at line 249 of file ng_btsocket.h.

Referenced by `ng_btsocket_l2cap_raw_control()`.

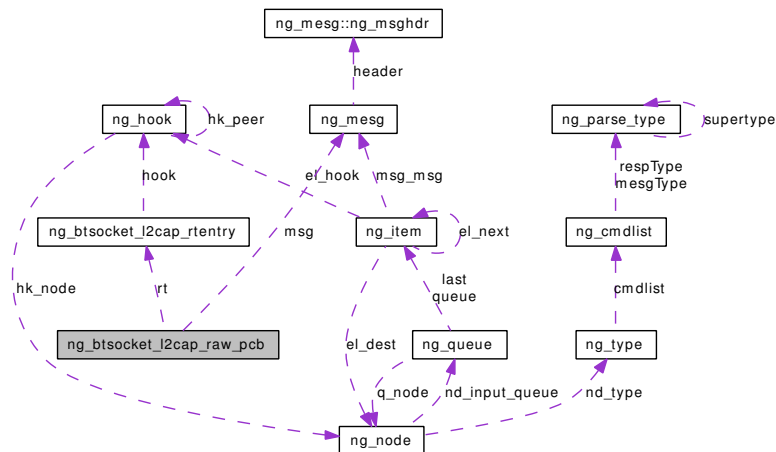
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.82 ng_btsocket_l2cap_raw_pcb Struct Reference

```
#include <ng_btsocket_l2cap.h>
```

Collaboration diagram for ng_btsocket_l2cap_raw_pcb:



Public Member Functions

- `LIST_ENTRY` (`ng_btsocket_l2cap_raw_pcb`) `next`

Data Fields

- `socket * so`
- `u_int32_t flags`
- `bdaddr_t src`
- `bdaddr_t dst`
- `ng_btsocket_l2cap_rtentry_p rt`
- `u_int32_t token`
- `ng_mesg * msg`
- `mtx pcb_mtx`

6.82.1 Detailed Description

Definition at line 65 of file `ng_btsocket_l2cap.h`.

6.82.2 Member Function Documentation

6.82.2.1 `ng_btsocket_l2cap_raw_pcb::LIST_ENTRY` (`ng_btsocket_l2cap_raw_pcb`)

6.82.3 Field Documentation

6.82.3.1 `bdaddr_t` `ng_btsocket_l2cap_raw_pcb::dst`

Definition at line 72 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, and `ng_btsocket_l2cap_raw_peeraddr()`.

6.82.3.2 `u_int32_t ng_btsocket_l2cap_raw_pcb::flags`

Definition at line 68 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.82.3.3 `struct ng_mesg* ng_btsocket_l2cap_raw_pcb::msg`

Definition at line 76 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, and `ng_btsocket_l2cap_raw_send_sync_ngmsg()`.

6.82.3.4 `struct mtx ng_btsocket_l2cap_raw_pcb::pcb_mtx`

Definition at line 78 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_detach()`, `ng_btsocket_l2cap_raw_disconnect()`, `ng_btsocket_l2cap_raw_input()`, `ng_btsocket_l2cap_raw_peeraddr()`, `ng_btsocket_l2cap_raw_rtclean()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, and `ng_btsocket_l2cap_raw_sockaddr()`.

6.82.3.5 `ng_btsocket_l2cap_rtenry_p ng_btsocket_l2cap_raw_pcb::rt`

Definition at line 73 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_disconnect()`, `ng_btsocket_l2cap_raw_rtclean()`, and `ng_btsocket_l2cap_raw_send_sync_ngmsg()`.

6.82.3.6 `struct socket* ng_btsocket_l2cap_raw_pcb::so`

Definition at line 66 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_rtclean()`.

6.82.3.7 `bdaddr_t ng_btsocket_l2cap_raw_pcb::src`

Definition at line 71 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, and `ng_btsocket_l2cap_raw_sockaddr()`.

6.82.3.8 `u_int32_t ng_btsocket_l2cap_raw_pcb::token`

Definition at line 75 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, and `ng_btsocket_l2cap_raw_send_sync_ngmsg()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h](#)

6.83 ng_btsocket_l2cap_raw_ping Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- `u_int32_t` [result](#)
- `u_int32_t` [echo_size](#)
- `u_int8_t *` [echo_data](#)

6.83.1 Detailed Description

Definition at line 227 of file `ng_btsocket.h`.

6.83.2 Field Documentation

6.83.2.1 `u_int8_t *` [ng_btsocket_l2cap_raw_ping::echo_data](#)

Definition at line 230 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.83.2.2 `u_int32_t` [ng_btsocket_l2cap_raw_ping::echo_size](#)

Definition at line 229 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

6.83.2.3 `u_int32_t` [ng_btsocket_l2cap_raw_ping::result](#)

Definition at line 228 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

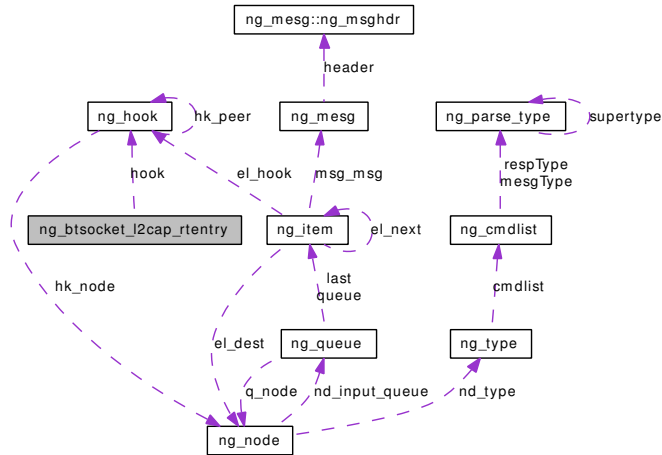
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.84 ng_btsocket_l2cap_rtenry Struct Reference

```
#include <ng_btsocket_l2cap.h>
```

Collaboration diagram for ng_btsocket_l2cap_rtenry:



Public Member Functions

- [LIST_ENTRY](#) ([ng_btsocket_l2cap_rtenry](#)) next

Data Fields

- [bdaddr_t](#) src
- [ng_hook](#) * hook

6.84.1 Detailed Description

Definition at line 44 of file `ng_btsocket_l2cap.h`.

6.84.2 Member Function Documentation

6.84.2.1 [ng_btsocket_l2cap_rtenry::LIST_ENTRY](#) ([ng_btsocket_l2cap_rtenry](#))

6.84.3 Field Documentation

6.84.3.1 `struct ng_hook* ng_btsocket_l2cap_rtenry::hook`

Definition at line 46 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, `ng_btsocket_l2cap_raw_rtclean()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_rtclean()`, `ng_btsocket_l2cap_send()`, `ng_btsocket_l2cap_send2()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, and `ng_btsocket_l2cap_send_l2ca_discon_req()`.

6.84.3.2 `bdaddr_t ng_btsocket_l2cap_rentry::src`

Definition at line 45 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, and `ng_btsocket_l2cap_raw_input()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h](#)

6.85 ng_btsocket_rfcomm_fc_info Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [u_int8_t lmodem](#)
- [u_int8_t rmodem](#)
- [u_int8_t tx_cred](#)
- [u_int8_t rx_cred](#)
- [u_int8_t cfc](#)
- [u_int8_t reserved](#)

6.85.1 Detailed Description

Definition at line 309 of file `ng_btsocket.h`.

6.85.2 Field Documentation

6.85.2.1 [u_int8_t ng_btsocket_rfcomm_fc_info::cfc](#)

Definition at line 314 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

6.85.2.2 [u_int8_t ng_btsocket_rfcomm_fc_info::lmodem](#)

Definition at line 310 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

6.85.2.3 [u_int8_t ng_btsocket_rfcomm_fc_info::reserved](#)

Definition at line 315 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

6.85.2.4 [u_int8_t ng_btsocket_rfcomm_fc_info::rmodem](#)

Definition at line 311 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

6.85.2.5 [u_int8_t ng_btsocket_rfcomm_fc_info::rx_cred](#)

Definition at line 313 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

6.85.2.6 `u_int8_t ng_btsocket_rfcomm_fc_info::tx_cred`

Definition at line 312 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`.

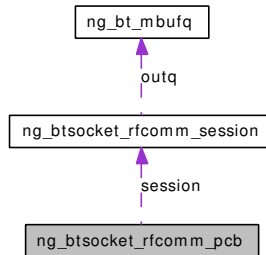
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.86 ng_btsocket_rfcomm_pcb Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Collaboration diagram for ng_btsocket_rfcomm_pcb:



Public Member Functions

- [LIST_ENTRY](#) ([ng_btsocket_rfcomm_pcb](#)) `session_next`
- [LIST_ENTRY](#) ([ng_btsocket_rfcomm_pcb](#)) `next`

Data Fields

- `socket * so`
- [ng_btsocket_rfcomm_session](#) * `session`
- `u_int16_t flags`
- `u_int16_t state`
- `bdaddr_t src`
- `bdaddr_t dst`
- `u_int8_t channel`
- `u_int8_t dlci`
- `u_int8_t lmodem`
- `u_int8_t rmodem`
- `u_int16_t mtu`
- `int16_t rx_cred`
- `int16_t tx_cred`
- `mtx pcb_mtx`
- `callout_handle timo`

6.86.1 Detailed Description

Definition at line 266 of file `ng_btsocket_rfcomm.h`.

6.86.2 Member Function Documentation

6.86.2.1 [ng_btsocket_rfcomm_pcb::LIST_ENTRY \(ng_btsocket_rfcomm_pcb\)](#)

6.86.2.2 [ng_btsocket_rfcomm_pcb::LIST_ENTRY \(ng_btsocket_rfcomm_pcb\)](#)

6.86.3 Field Documentation

6.86.3.1 [u_int8_t ng_btsocket_rfcomm_pcb::channel](#)

Definition at line 288 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_bind\(\)](#), [ng_btsocket_rfcomm_connect\(\)](#), [ng_btsocket_rfcomm_listen\(\)](#), [ng_btsocket_rfcomm_pcb_by_channel\(\)](#), [ng_btsocket_rfcomm_pcb_listener\(\)](#), [ng_btsocket_rfcomm_peeraddr\(\)](#), and [ng_btsocket_rfcomm_sockaddr\(\)](#).

6.86.3.2 [u_int8_t ng_btsocket_rfcomm_pcb::dlci](#)

Definition at line 289 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_connect\(\)](#), [ng_btsocket_rfcomm_pcb_by_dlci\(\)](#), [ng_btsocket_rfcomm_pcb_kill\(\)](#), [ng_btsocket_rfcomm_pcb_send\(\)](#), [ng_btsocket_rfcomm_process_timeout\(\)](#), [ng_btsocket_rfcomm_receive_pn\(\)](#), [ng_btsocket_rfcomm_receive_sabm\(\)](#), [ng_btsocket_rfcomm_send_credits\(\)](#), [ng_btsocket_rfcomm_send_msc\(\)](#), [ng_btsocket_rfcomm_send_pn\(\)](#), [ng_btsocket_rfcomm_session_clean\(\)](#), and [ng_btsocket_rfcomm_set_pn\(\)](#).

6.86.3.3 [bdaddr_t ng_btsocket_rfcomm_pcb::dst](#)

Definition at line 286 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_connect\(\)](#), and [ng_btsocket_rfcomm_peeraddr\(\)](#).

6.86.3.4 [u_int16_t ng_btsocket_rfcomm_pcb::flags](#)

Definition at line 270 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_ctloutput\(\)](#), [ng_btsocket_rfcomm_detach\(\)](#), [ng_btsocket_rfcomm_disconnect\(\)](#), [ng_btsocket_rfcomm_pcb_kill\(\)](#), [ng_btsocket_rfcomm_pcb_send\(\)](#), [ng_btsocket_rfcomm_process_timeout\(\)](#), [ng_btsocket_rfcomm_receive_disc\(\)](#), [ng_btsocket_rfcomm_receive_dm\(\)](#), [ng_btsocket_rfcomm_receive_msc\(\)](#), [ng_btsocket_rfcomm_receive_pn\(\)](#), [ng_btsocket_rfcomm_receive_sabm\(\)](#), [ng_btsocket_rfcomm_receive_ua\(\)](#), [ng_btsocket_rfcomm_receive_uih\(\)](#), [ng_btsocket_rfcomm_send\(\)](#), [ng_btsocket_rfcomm_send_credits\(\)](#), [ng_btsocket_rfcomm_send_msc\(\)](#), [ng_btsocket_rfcomm_send_pn\(\)](#), [ng_btsocket_rfcomm_session_clean\(\)](#), [ng_btsocket_rfcomm_session_process_pcb\(\)](#), [ng_btsocket_rfcomm_set_pn\(\)](#), [ng_btsocket_rfcomm_timeout\(\)](#), and [ng_btsocket_rfcomm_untimeout\(\)](#).

6.86.3.5 [u_int8_t ng_btsocket_rfcomm_pcb::lmodem](#)

Definition at line 291 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_ctloutput\(\)](#), [ng_btsocket_rfcomm_receive_uih\(\)](#), and [ng_btsocket_rfcomm_send_msc\(\)](#).

6.86.3.6 `u_int16_t ng_btsocket_rfcomm_pcb::mtu`

Definition at line 294 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_pn()`, and `ng_btsocket_rfcomm_set_pn()`.

6.86.3.7 `struct mtx ng_btsocket_rfcomm_pcb::pcb_mtx`

Definition at line 298 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_disconnect()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, `ng_btsocket_rfcomm_send_pn()`, `ng_btsocket_rfcomm_session_clean()`, `ng_btsocket_rfcomm_session_process_pcb()`, `ng_btsocket_rfcomm_set_pn()`, `ng_btsocket_rfcomm_timeout()`, and `ng_btsocket_rfcomm_untimeout()`.

6.86.3.8 `u_int8_t ng_btsocket_rfcomm_pcb::rmodem`

Definition at line 292 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_pcb_send()`, and `ng_btsocket_rfcomm_receive_msc()`.

6.86.3.9 `int16_t ng_btsocket_rfcomm_pcb::rx_cred`

Definition at line 295 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_pn()`, and `ng_btsocket_rfcomm_set_pn()`.

6.86.3.10 `struct ng_btsocket_rfcomm_session* ng_btsocket_rfcomm_pcb::session`

Definition at line 268 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, and `ng_btsocket_rfcomm_send_pn()`.

6.86.3.11 `struct socket* ng_btsocket_rfcomm_pcb::so`

Definition at line 267 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb_listener()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send()`, and `ng_btsocket_rfcomm_send_credits()`.

6.86.3.12 `bdaddr_t ng_btsocket_rfcomm_pcb::src`

Definition at line 285 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_bind()`, `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_pcb_by_channel()`, `ng_btsocket_rfcomm_pcb_listener()`, and `ng_btsocket_rfcomm_sockaddr()`.

6.86.3.13 `u_int16_t ng_btsocket_rfcomm_pcb::state`

Definition at line 277 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_disconnect()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, `ng_btsocket_rfcomm_send_pn()`, `ng_btsocket_rfcomm_session_clean()`, `ng_btsocket_rfcomm_session_process_pcb()`, and `ng_btsocket_rfcomm_set_pn()`.

6.86.3.14 `struct callout_handle ng_btsocket_rfcomm_pcb::timo`

Definition at line 299 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_timeout()`, and `ng_btsocket_rfcomm_untimeout()`.

6.86.3.15 `int16_t ng_btsocket_rfcomm_pcb::tx_cred`

Definition at line 296 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_credits()`, and `ng_btsocket_rfcomm_set_pn()`.

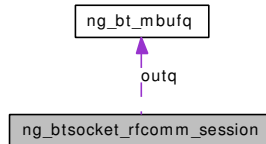
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h`

6.87 ng_btsocket_rfcomm_session Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Collaboration diagram for ng_btsocket_rfcomm_session:



Public Member Functions

- [LIST_HEAD](#) ([, ng_btsocket_rfcomm_pcb](#)) [dlcs](#)
- [LIST_ENTRY](#) ([ng_btsocket_rfcomm_session](#)) [next](#)

Data Fields

- socket * [l2so](#)
- [u_int16_t](#) [state](#)
- [u_int16_t](#) [flags](#)
- [u_int16_t](#) [mtu](#)
- [ng_bt_mbufq](#) [outq](#)
- [mtx](#) [session_mtx](#)

6.87.1 Detailed Description

Definition at line 232 of file `ng_btsocket_rfcomm.h`.

6.87.2 Member Function Documentation

6.87.2.1 [ng_btsocket_rfcomm_session::LIST_ENTRY](#) ([ng_btsocket_rfcomm_session](#))

6.87.2.2 [ng_btsocket_rfcomm_session::LIST_HEAD](#) ([ng_btsocket_rfcomm_pcb](#))

6.87.3 Field Documentation

6.87.3.1 [u_int16_t](#) [ng_btsocket_rfcomm_session::flags](#)

Definition at line 243 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_send_uih()`, `ng_btsocket_rfcomm_session_accept()`, `ng_btsocket_rfcomm_session_connect()`, `ng_btsocket_rfcomm_session_receive()`, `ng_btsocket_rfcomm_session_send()`, and `ng_btsocket_rfcomm_session_task()`.

6.87.3.2 `struct socket*` `ng_btsocket_rfcomm_session::l2so`

Definition at line 233 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_session_accept()`, `ng_btsocket_rfcomm_session_by_addr()`, `ng_btsocket_rfcomm_session_connect()`, `ng_btsocket_rfcomm_session_receive()`, `ng_btsocket_rfcomm_session_send()`, and `ng_btsocket_rfcomm_session_task()`.

6.87.3.3 `u_int16_t` `ng_btsocket_rfcomm_session::mtu`

Definition at line 251 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_session_accept()`, and `ng_btsocket_rfcomm_session_connect()`.

6.87.3.4 `struct ng_bt_mbufq` `ng_btsocket_rfcomm_session::outq`

Definition at line 252 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_send_uih()`, and `ng_btsocket_rfcomm_session_send()`.

6.87.3.5 `struct mtx` `ng_btsocket_rfcomm_session::session_mtx`

Definition at line 254 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_pcb_by_dlci()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_frame()`, `ng_btsocket_rfcomm_receive_mcc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, `ng_btsocket_rfcomm_send_pn()`, `ng_btsocket_rfcomm_send_uih()`, `ng_btsocket_rfcomm_session_accept()`, `ng_btsocket_rfcomm_session_clean()`, `ng_btsocket_rfcomm_session_connect()`, `ng_btsocket_rfcomm_session_process_pcb()`, `ng_btsocket_rfcomm_session_receive()`, `ng_btsocket_rfcomm_session_send()`, `ng_btsocket_rfcomm_session_task()`, and `ng_btsocket_rfcomm_sessions_task()`.

6.87.3.6 `u_int16_t` `ng_btsocket_rfcomm_session::state`

Definition at line 235 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_listen()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_`

sabm(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_receive_uih(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_uih(), ng_btsocket_rfcomm_session_accept(), ng_btsocket_rfcomm_session_connect(), ng_btsocket_rfcomm_session_receive(), ng_btsocket_rfcomm_session_send(), and ng_btsocket_rfcomm_session_task().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.88 ng_cisco_ipaddr Struct Reference

```
#include <ng_cisco.h>
```

Data Fields

- in_addr [ipaddr](#)
- in_addr [netmask](#)

6.88.1 Detailed Description

Definition at line 66 of file ng_cisco.h.

6.88.2 Field Documentation

6.88.2.1 struct in_addr [ng_cisco_ipaddr::ipaddr](#)

Definition at line 67 of file ng_cisco.h.

6.88.2.2 struct in_addr [ng_cisco_ipaddr::netmask](#)

Definition at line 68 of file ng_cisco.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_cisco.h](#)

6.89 ng_cisco_stats Struct Reference

```
#include <ng_cisco.h>
```

Data Fields

- `u_int32_t seqRetries`
- `u_int32_t keepAlivePeriod`

6.89.1 Detailed Description

Definition at line 78 of file `ng_cisco.h`.

6.89.2 Field Documentation

6.89.2.1 `u_int32_t ng_cisco_stats::keepAlivePeriod`

Definition at line 80 of file `ng_cisco.h`.

6.89.2.2 `u_int32_t ng_cisco_stats::seqRetries`

Definition at line 79 of file `ng_cisco.h`.

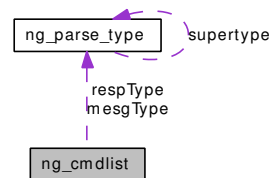
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_cisco.h`

6.90 ng_cmdlist Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_cmdlist:



Data Fields

- `u_int32_t` [cookie](#)
- `int` [cmd](#)
- `const char *` [name](#)
- `ng_parse_type *` [mesgType](#)
- `ng_parse_type *` [respType](#)

6.90.1 Detailed Description

Definition at line 990 of file netgraph.h.

6.90.2 Field Documentation

6.90.2.1 `int` [ng_cmdlist::cmd](#)

Definition at line 992 of file netgraph.h.

Referenced by [ng_generic_msg\(\)](#).

6.90.2.2 `u_int32_t` [ng_cmdlist::cookie](#)

Definition at line 991 of file netgraph.h.

Referenced by [ng_generic_msg\(\)](#).

6.90.2.3 `struct ng_parse_type*` [ng_cmdlist::mesgType](#)

Definition at line 994 of file netgraph.h.

6.90.2.4 `const char*` [ng_cmdlist::name](#)

Definition at line 993 of file netgraph.h.

Referenced by [ng_generic_msg\(\)](#).

6.90.2.5 struct [ng_parse_type*](#) [ng_cmdlist::respType](#)

Definition at line 995 of file netgraph.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netgraph.h](#)

6.91 ng_deflate_config Struct Reference

```
#include <ng_deflate.h>
```

Data Fields

- u_char [enable](#)
- u_char [windowBits](#)

6.91.1 Detailed Description

Definition at line 42 of file ng_deflate.h.

6.91.2 Field Documentation

6.91.2.1 u_char [ng_deflate_config::enable](#)

Definition at line 43 of file ng_deflate.h.

6.91.2.2 u_char [ng_deflate_config::windowBits](#)

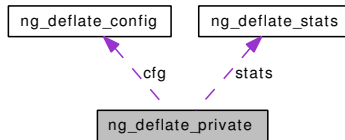
Definition at line 44 of file ng_deflate.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_deflate.h](#)

6.92 ng_deflate_private Struct Reference

Collaboration diagram for ng_deflate_private:



Data Fields

- [ng_deflate_config](#) `cfg`
- `u_char` [inbuf](#) [DEFLATE_BUF_SIZE]
- `u_char` [outbuf](#) [DEFLATE_BUF_SIZE]
- `z_stream` `cx`
- [ng_deflate_stats](#) `stats`
- `ng_ID_t` [ctrlnode](#)
- `uint16_t` [seqnum](#)
- `u_char` [compress](#)

6.92.1 Detailed Description

Definition at line 61 of file `ng_deflate.c`.

6.92.2 Field Documentation

6.92.2.1 struct [ng_deflate_config](#) `ng_deflate_private::cfg`

Definition at line 62 of file `ng_deflate.c`.

6.92.2.2 `u_char` [ng_deflate_private::compress](#)

Definition at line 69 of file `ng_deflate.c`.

6.92.2.3 `ng_ID_t` [ng_deflate_private::ctrlnode](#)

Definition at line 67 of file `ng_deflate.c`.

6.92.2.4 `z_stream` [ng_deflate_private::cx](#)

Definition at line 65 of file `ng_deflate.c`.

6.92.2.5 `u_char` [ng_deflate_private::inbuf](#)[DEFLATE_BUF_SIZE]

Definition at line 63 of file `ng_deflate.c`.

6.92.2.6 `u_char ng_deflate_private::outbuf[DEFLATE_BUF_SIZE]`

Definition at line 64 of file `ng_deflate.c`.

6.92.2.7 `uint16_t ng_deflate_private::seqnum`

Definition at line 68 of file `ng_deflate.c`.

6.92.2.8 `struct ng_deflate_stats ng_deflate_private::stats`

Definition at line 66 of file `ng_deflate.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_deflate.c](#)

6.93 ng_deflate_stats Struct Reference

```
#include <ng_deflate.h>
```

Data Fields

- uint64_t [FramesPlain](#)
- uint64_t [FramesComp](#)
- uint64_t [FramesUncomp](#)
- uint64_t [InOctets](#)
- uint64_t [OutOctets](#)
- uint64_t [Errors](#)

6.93.1 Detailed Description

Definition at line 55 of file `ng_deflate.h`.

6.93.2 Field Documentation

6.93.2.1 uint64_t [ng_deflate_stats::Errors](#)

Definition at line 61 of file `ng_deflate.h`.

6.93.2.2 uint64_t [ng_deflate_stats::FramesComp](#)

Definition at line 57 of file `ng_deflate.h`.

6.93.2.3 uint64_t [ng_deflate_stats::FramesPlain](#)

Definition at line 56 of file `ng_deflate.h`.

6.93.2.4 uint64_t [ng_deflate_stats::FramesUncomp](#)

Definition at line 58 of file `ng_deflate.h`.

6.93.2.5 uint64_t [ng_deflate_stats::InOctets](#)

Definition at line 59 of file `ng_deflate.h`.

6.93.2.6 uint64_t [ng_deflate_stats::OutOctets](#)

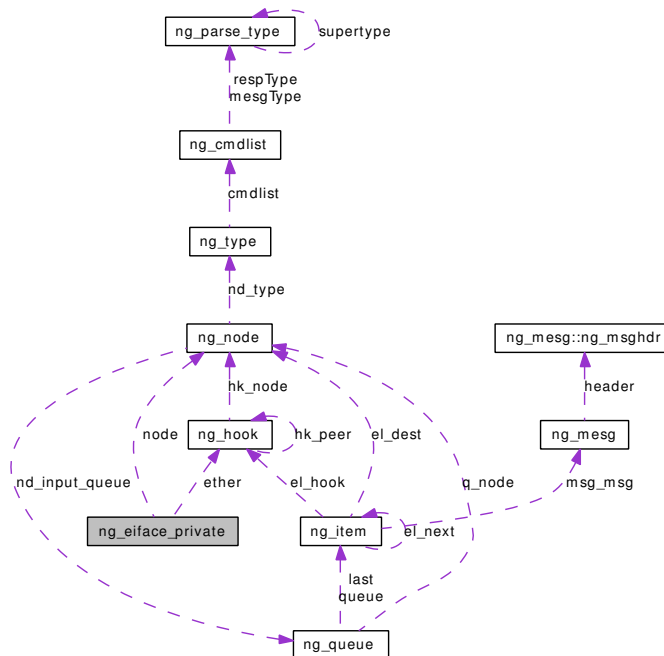
Definition at line 60 of file `ng_deflate.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_deflate.h](#)

6.94 ng_iface_private Struct Reference

Collaboration diagram for ng_iface_private:



Data Fields

- `ifnet *` `ifp`
- `int` `unit`
- `node_p` `node`
- `hook_p` `ether`

6.94.1 Detailed Description

Definition at line 74 of file `ng_iface.c`.

6.94.2 Field Documentation

6.94.2.1 `hook_p ng_iface_private::ether`

Definition at line 78 of file `ng_iface.c`.

6.94.2.2 `struct ifnet* ng_iface_private::ifp`

Definition at line 75 of file `ng_iface.c`.

6.94.2.3 `node_p ng_eiface_private::node`

Definition at line 77 of file `ng_eiface.c`.

6.94.2.4 `int ng_eiface_private::unit`

Definition at line 76 of file `ng_eiface.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_eiface.c](#)

6.95 ng_etfilter Struct Reference

```
#include <ng_etf.h>
```

Data Fields

- char [matchhook](#) [NG_HOOKSIZ]
- u_int16_t [ethertype](#)

6.95.1 Detailed Description

Definition at line 76 of file ng_etf.h.

6.95.2 Field Documentation

6.95.2.1 u_int16_t [ng_etfilter::ethertype](#)

Definition at line 78 of file ng_etf.h.

6.95.2.2 char [ng_etfilter::matchhook](#)[NG_HOOKSIZ]

Definition at line 77 of file ng_etf.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/[ng_etf.h](#)

6.96 ng_etfstat Struct Reference

```
#include <ng_etf.h>
```

Data Fields

- [u_int32_t packets_in](#)
- [u_int32_t packets_out](#)

6.96.1 Detailed Description

Definition at line 61 of file [ng_etf.h](#).

6.96.2 Field Documentation

6.96.2.1 [u_int32_t ng_etfstat::packets_in](#)

Definition at line 62 of file [ng_etf.h](#).

6.96.2.2 [u_int32_t ng_etfstat::packets_out](#)

Definition at line 63 of file [ng_etf.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_etf.h](#)

6.97 ng_fec_bundle Struct Reference

6.97.1 Detailed Description

Definition at line 155 of file ng_fec.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_fec.c](#)

6.98 ng_fec_ifname Struct Reference

```
#include <ng_fec.h>
```

Data Fields

- char [ngif_name](#) [NG_FEC_FEC_NAME_MAX+1]

6.98.1 Detailed Description

Definition at line 109 of file [ng_fec.h](#).

6.98.2 Field Documentation

6.98.2.1 char [ng_fec_ifname::ngif_name](#)[NG_FEC_FEC_NAME_MAX+1]

Definition at line 110 of file [ng_fec.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_fec.h](#)

6.99 ng_fec_portlist Struct Reference

Data Fields

- ifnet * [fec_if](#)
- void(* [fec_if_input](#))(struct ifnet *, struct mbuf *)
- int [fec_idx](#)
- int [fec_ifstat](#)
- ether_addr [fec_mac](#)

6.99.1 Detailed Description

Definition at line 145 of file ng_fec.c.

6.99.2 Field Documentation

6.99.2.1 int [ng_fec_portlist::fec_idx](#)

Definition at line 149 of file ng_fec.c.

Referenced by [ng_fec_choose_port\(\)](#).

6.99.2.2 struct ifnet* [ng_fec_portlist::fec_if](#)

Definition at line 146 of file ng_fec.c.

Referenced by [ng_fec_addport\(\)](#), [ng_fec_choose_port\(\)](#), [ng_fec_delport\(\)](#), [ng_fec_init\(\)](#), [ng_fec_input\(\)](#), [ng_fec_ioctl\(\)](#), [ng_fec_setport\(\)](#), [ng_fec_shutdown\(\)](#), [ng_fec_stop\(\)](#), and [ng_fec_tick\(\)](#).

6.99.2.3 void(* [ng_fec_portlist::fec_if_input](#))(struct ifnet *, struct mbuf *)

Referenced by [ng_fec_delport\(\)](#).

6.99.2.4 int [ng_fec_portlist::fec_ifstat](#)

Definition at line 150 of file ng_fec.c.

Referenced by [ng_fec_choose_port\(\)](#), [ng_fec_ifmedia_sts\(\)](#), [ng_fec_init\(\)](#), and [ng_fec_tick\(\)](#).

6.99.2.5 struct ether_addr [ng_fec_portlist::fec_mac](#)

Definition at line 151 of file ng_fec.c.

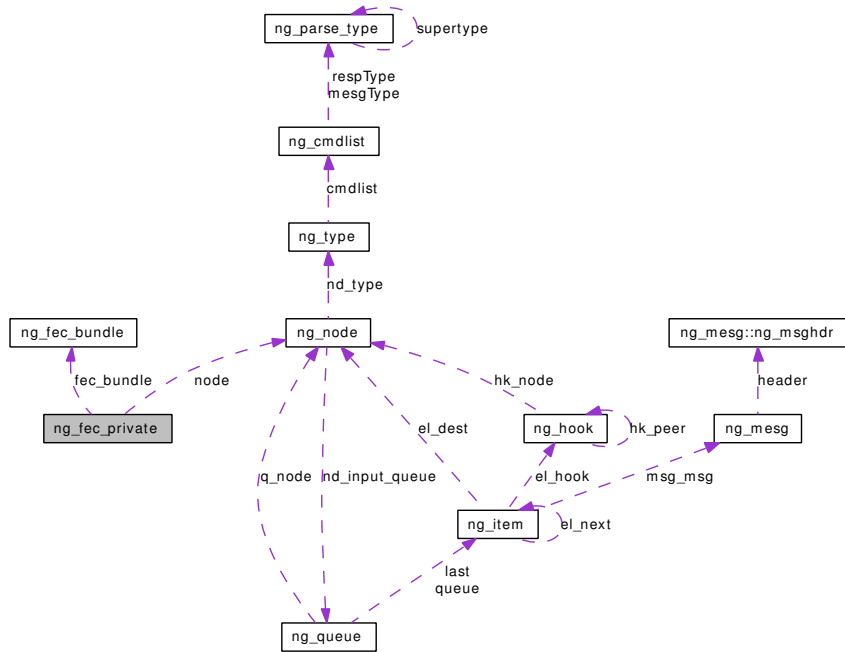
Referenced by [ng_fec_delport\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_fec.c](#)

6.100 ng_fec_private Struct Reference

Collaboration diagram for ng_fec_private:



Data Fields

- ifnet * ifp
- ifmedia ifmedia
- int if_flags
- int if_error
- int unit
- node_p node
- ng_fec_bundle fec_bundle
- callout_handle fec_ch

6.100.1 Detailed Description

Definition at line 170 of file ng_fec.c.

6.100.2 Field Documentation

6.100.2.1 struct ng_fec_bundle ng_fec_private::fec_bundle

Definition at line 177 of file ng_fec.c.

Referenced by ng_fec_addport(), ng_fec_delport(), ng_fec_ifmedia_sts(), ng_fec_init(), ng_fec_input(), ng_fec_ioctl(), ng_fec_setport(), ng_fec_start(), ng_fec_stop(), and ng_fec_tick().

6.100.2.2 struct callout_handle [ng_fec_private::fec_ch](#)

Definition at line 178 of file [ng_fec.c](#).

Referenced by [ng_fec_init\(\)](#), [ng_fec_stop\(\)](#), and [ng_fec_tick\(\)](#).

6.100.2.3 int [ng_fec_private::if_error](#)

Definition at line 174 of file [ng_fec.c](#).

Referenced by [ng_fec_start\(\)](#).

6.100.2.4 int [ng_fec_private::if_flags](#)

Definition at line 173 of file [ng_fec.c](#).

Referenced by [ng_fec_ioctl\(\)](#).

6.100.2.5 struct ifmedia [ng_fec_private::ifmedia](#)

Definition at line 172 of file [ng_fec.c](#).

Referenced by [ng_fec_ioctl\(\)](#).

6.100.2.6 struct ifnet* [ng_fec_private::ifp](#)

Definition at line 171 of file [ng_fec.c](#).

Referenced by [ng_fec_addport\(\)](#), [ng_fec_delpport\(\)](#), [ng_fec_init\(\)](#), [ng_fec_input\(\)](#), and [ng_fec_tick\(\)](#).

6.100.2.7 node_p [ng_fec_private::node](#)

Definition at line 176 of file [ng_fec.c](#).

Referenced by [ng_fec_addport\(\)](#).

6.100.2.8 int [ng_fec_private::unit](#)

Definition at line 175 of file [ng_fec.c](#).

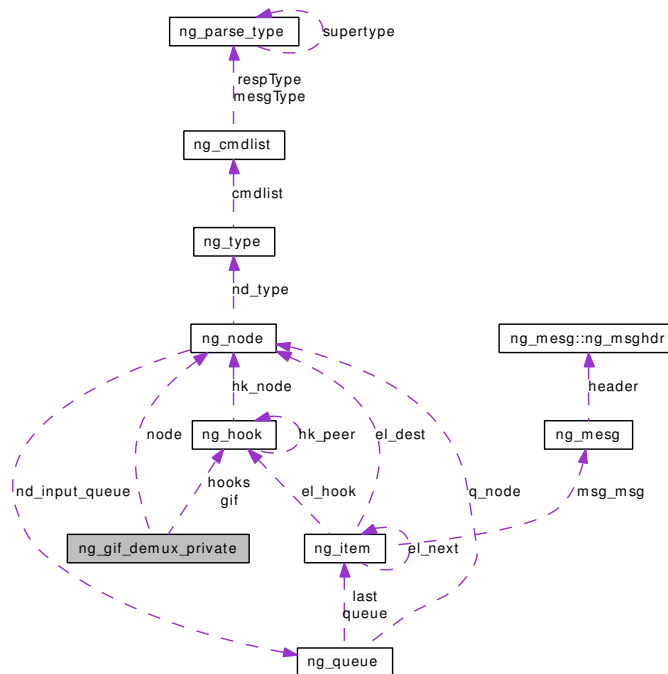
Referenced by [ng_fec_addport\(\)](#), [ng_fec_delpport\(\)](#), [ng_fec_init\(\)](#), [ng_fec_ioctl\(\)](#), and [ng_fec_tick\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_fec.c](#)

6.101 ng_gif_demux_private Struct Reference

Collaboration diagram for ng_gif_demux_private:



Data Fields

- `node_p` node
- `hook_p` gif
- `hook_p` hooks [NUM_FAMILIES]

6.101.1 Detailed Description

Definition at line 117 of file `ng_gif_demux.c`.

6.101.2 Field Documentation

6.101.2.1 hook_p ng_gif_demux_private::gif

Definition at line 119 of file `ng_gif_demux.c`.

6.101.2.2 hook_p ng_gif_demux_private::hooks[NUM_FAMILIES]

Definition at line 120 of file `ng_gif_demux.c`.

6.101.2.3 `node_p ng_gif_demux_private::node`

Definition at line 118 of file `ng_gif_demux.c`.

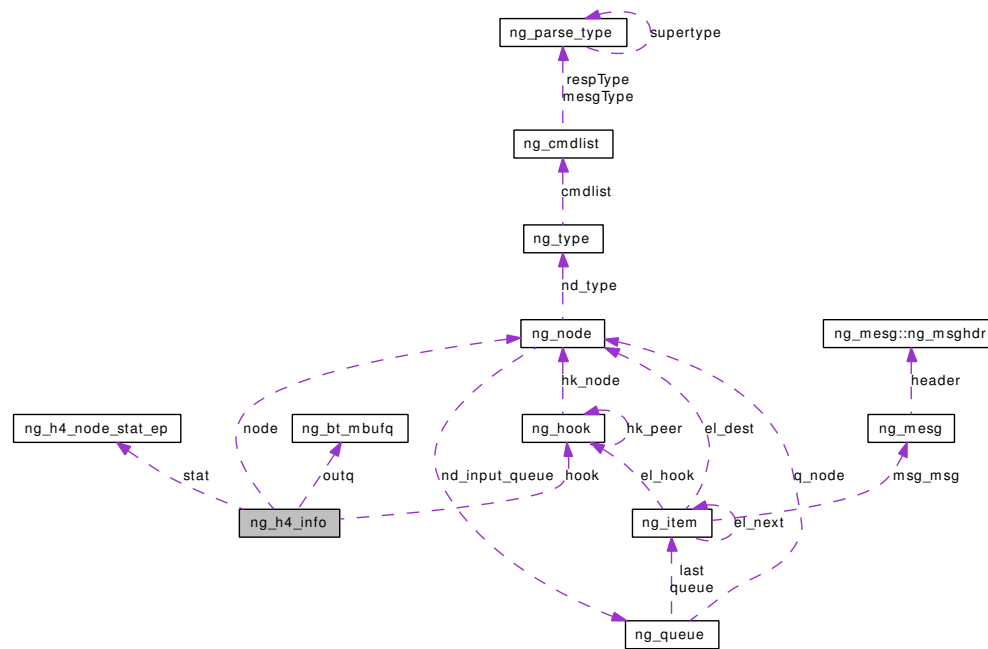
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_gif_demux.c`

6.102 ng_h4_info Struct Reference

```
#include <ng_h4_var.h>
```

Collaboration diagram for ng_h4_info:



Data Fields

- `tty * tp`
- `node_p node`
- `u_int32_t flags`
- `ng_h4_node_debug_ep debug`
- `ng_h4_node_state_ep state`
- `ng_h4_node_stat_ep stat`
- `ng_bt_mbufq_t outq`
- `u_int8_t ibuf [NG_H4_IBUF_SIZE]`
- `u_int32_t got`
- `u_int32_t want`
- `hook_p hook`
- `callout timo`

6.102.1 Detailed Description

Definition at line 68 of file `ng_h4_var.h`.

6.102.2 Field Documentation

6.102.2.1 `ng_h4_node_debug_ep ng_h4_info::debug`

Definition at line 75 of file `ng_h4_var.h`.

Referenced by `ng_h4_rcvmsg()`.

6.102.2.2 `u_int32_t ng_h4_info::flags`

Definition at line 72 of file `ng_h4_var.h`.

Referenced by `ng_h4_close()`, `ng_h4_disconnect()`, `ng_h4_process_timeout()`, `ng_h4_rcvmsg()`, `ng_h4_start2()`, `ng_h4_timeout()`, and `ng_h4_untimeout()`.

6.102.2.3 `u_int32_t ng_h4_info::got`

Definition at line 93 of file `ng_h4_var.h`.

Referenced by `ng_h4_disconnect()`, `ng_h4_input()`, and `ng_h4_rcvmsg()`.

6.102.2.4 `hook_p ng_h4_info::hook`

Definition at line 96 of file `ng_h4_var.h`.

Referenced by `ng_h4_connect()`, `ng_h4_disconnect()`, `ng_h4_input()`, `ng_h4_newhook()`, `ng_h4_rcvdata()`, and `ng_h4_rcvmsg()`.

6.102.2.5 `u_int8_t ng_h4_info::ibuf[NG_H4_IBUF_SIZE]`

Definition at line 92 of file `ng_h4_var.h`.

Referenced by `ng_h4_input()`.

6.102.2.6 `node_p ng_h4_info::node`

Definition at line 70 of file `ng_h4_var.h`.

Referenced by `ng_h4_close()`, `ng_h4_input()`, `ng_h4_ioctl()`, `ng_h4_rcvdata()`, `ng_h4_shutdown()`, and `ng_h4_start()`.

6.102.2.7 `ng_bt_mbufq_t ng_h4_info::outq`

Definition at line 87 of file `ng_h4_var.h`.

Referenced by `ng_h4_close()`, `ng_h4_disconnect()`, `ng_h4_rcvdata()`, `ng_h4_rcvmsg()`, and `ng_h4_start2()`.

6.102.2.8 `ng_h4_node_stat_ep ng_h4_info::stat`

Definition at line 78 of file `ng_h4_var.h`.

Referenced by `ng_h4_input()`, `ng_h4_rcvdata()`, `ng_h4_rcvmsg()`, and `ng_h4_start2()`.

6.102.2.9 `ng_h4_node_state_ep ng_h4_info::state`

Definition at line 76 of file `ng_h4_var.h`.

Referenced by `ng_h4_disconnect()`, `ng_h4_input()`, and `ng_h4_rcvmsg()`.

6.102.2.10 `struct callout ng_h4_info::timo`

Definition at line 97 of file `ng_h4_var.h`.

Referenced by `ng_h4_timeout()`, and `ng_h4_untimeout()`.

6.102.2.11 `struct tty* ng_h4_info::tp`

Definition at line 69 of file `ng_h4_var.h`.

Referenced by `ng_h4_input()`, `ng_h4_start()`, and `ng_h4_start2()`.

6.102.2.12 `u_int32_t ng_h4_info::want`

Definition at line 94 of file `ng_h4_var.h`.

Referenced by `ng_h4_disconnect()`, `ng_h4_input()`, and `ng_h4_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_var.h`

6.103 ng_h4_node_stat_ep Struct Reference

```
#include <ng_h4.h>
```

Data Fields

- [u_int32_t pckts_recv](#)
- [u_int32_t bytes_recv](#)
- [u_int32_t pckts_sent](#)
- [u_int32_t bytes_sent](#)
- [u_int32_t oerrors](#)
- [u_int32_t ierrors](#)

6.103.1 Detailed Description

Definition at line 100 of file ng_h4.h.

6.103.2 Field Documentation

6.103.2.1 [u_int32_t ng_h4_node_stat_ep::bytes_recv](#)

Definition at line 102 of file ng_h4.h.

6.103.2.2 [u_int32_t ng_h4_node_stat_ep::bytes_sent](#)

Definition at line 104 of file ng_h4.h.

6.103.2.3 [u_int32_t ng_h4_node_stat_ep::ierrors](#)

Definition at line 106 of file ng_h4.h.

6.103.2.4 [u_int32_t ng_h4_node_stat_ep::oerrors](#)

Definition at line 105 of file ng_h4.h.

6.103.2.5 [u_int32_t ng_h4_node_stat_ep::pckts_recv](#)

Definition at line 101 of file ng_h4.h.

6.103.2.6 [u_int32_t ng_h4_node_stat_ep::pckts_sent](#)

Definition at line 103 of file ng_h4.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_h4.h](#)

6.104 ng_hci_lp_con_cfm_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int8_t status](#)
- [u_int8_t link_type](#)
- [u_int16_t con_handle](#)
- [bdaddr_t bdaddr](#)

6.104.1 Detailed Description

Definition at line 414 of file `ng_hci.h`.

6.104.2 Field Documentation

6.104.2.1 [bdaddr_t ng_hci_lp_con_cfm_ep::bdaddr](#)

Definition at line 418 of file `ng_hci.h`.

6.104.2.2 [u_int16_t ng_hci_lp_con_cfm_ep::con_handle](#)

Definition at line 417 of file `ng_hci.h`.

6.104.2.3 [u_int8_t ng_hci_lp_con_cfm_ep::link_type](#)

Definition at line 416 of file `ng_hci.h`.

6.104.2.4 [u_int8_t ng_hci_lp_con_cfm_ep::status](#)

Definition at line 415 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.105 ng_hci_lp_con_ind_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int8_t link_type`
- `u_int8_t uclass [NG_HCI_CLASS_SIZE]`
- `bdaddr_t bdaddr`

6.105.1 Detailed Description

Definition at line 423 of file `ng_hci.h`.

6.105.2 Field Documentation

6.105.2.1 `bdaddr_t ng_hci_lp_con_ind_ep::bdaddr`

Definition at line 426 of file `ng_hci.h`.

6.105.2.2 `u_int8_t ng_hci_lp_con_ind_ep::link_type`

Definition at line 424 of file `ng_hci.h`.

6.105.2.3 `u_int8_t ng_hci_lp_con_ind_ep::uclass[NG_HCI_CLASS_SIZE]`

Definition at line 425 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.106 ng_hci_lp_con_req_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t link_type](#)
- [bdaddr_t bdaddr](#)

6.106.1 Detailed Description

Definition at line 395 of file `ng_hci.h`.

6.106.2 Field Documentation

6.106.2.1 [bdaddr_t ng_hci_lp_con_req_ep::bdaddr](#)

Definition at line 397 of file `ng_hci.h`.

Referenced by `ng_hci_lp_acl_con_req()`, and `ng_hci_lp_sco_con_req()`.

6.106.2.2 [u_int16_t ng_hci_lp_con_req_ep::link_type](#)

Definition at line 396 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.107 ng_hci_lp_con_rsp_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int8_t status](#)
- [u_int8_t link_type](#)
- [bdaddr_t bdaddr](#)

6.107.1 Detailed Description

Definition at line 431 of file `ng_hci.h`.

6.107.2 Field Documentation

6.107.2.1 [bdaddr_t ng_hci_lp_con_rsp_ep::bdaddr](#)

Definition at line 434 of file `ng_hci.h`.

6.107.2.2 [u_int8_t ng_hci_lp_con_rsp_ep::link_type](#)

Definition at line 433 of file `ng_hci.h`.

6.107.2.3 [u_int8_t ng_hci_lp_con_rsp_ep::status](#)

Definition at line 432 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.108 ng_hci_lp_discon_ind_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int8_t reason](#)
- [u_int8_t link_type](#)
- [u_int16_t con_handle](#)

6.108.1 Detailed Description

Definition at line 439 of file `ng_hci.h`.

6.108.2 Field Documentation

6.108.2.1 [u_int16_t ng_hci_lp_discon_ind_ep::con_handle](#)

Definition at line 442 of file `ng_hci.h`.

6.108.2.2 [u_int8_t ng_hci_lp_discon_ind_ep::link_type](#)

Definition at line 441 of file `ng_hci.h`.

6.108.2.3 [u_int8_t ng_hci_lp_discon_ind_ep::reason](#)

Definition at line 440 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.109 ng_hci_lp_discon_req_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t con_handle](#)
- [u_int16_t reason](#)

6.109.1 Detailed Description

Definition at line 407 of file ng_hci.h.

6.109.2 Field Documentation

6.109.2.1 [u_int16_t ng_hci_lp_discon_req_ep::con_handle](#)

Definition at line 408 of file ng_hci.h.

6.109.2.2 [u_int16_t ng_hci_lp_discon_req_ep::reason](#)

Definition at line 409 of file ng_hci.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_hci.h](#)

6.110 ng_hci_lp_qos_cfm_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int16_t` [status](#)
- `u_int16_t` [con_handle](#)

6.110.1 Detailed Description

Definition at line 459 of file `ng_hci.h`.

6.110.2 Field Documentation

6.110.2.1 `u_int16_t` [ng_hci_lp_qos_cfm_ep::con_handle](#)

Definition at line 461 of file `ng_hci.h`.

6.110.2.2 `u_int16_t` [ng_hci_lp_qos_cfm_ep::status](#)

Definition at line 460 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.111 ng_hci_lp_qos_ind_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t con_handle](#)

6.111.1 Detailed Description

Definition at line 466 of file `ng_hci.h`.

6.111.2 Field Documentation

6.111.2.1 [u_int16_t ng_hci_lp_qos_ind_ep::con_handle](#)

Definition at line 467 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_hci.h](#)

6.112 ng_hci_lp_qos_req_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t con_handle](#)
- [u_int8_t flags](#)
- [u_int8_t service_type](#)
- [u_int32_t token_rate](#)
- [u_int32_t peak_bandwidth](#)
- [u_int32_t latency](#)
- [u_int32_t delay_variation](#)

6.112.1 Detailed Description

Definition at line 447 of file ng_hci.h.

6.112.2 Field Documentation

6.112.2.1 [u_int16_t ng_hci_lp_qos_req_ep::con_handle](#)

Definition at line 448 of file ng_hci.h.

6.112.2.2 [u_int32_t ng_hci_lp_qos_req_ep::delay_variation](#)

Definition at line 454 of file ng_hci.h.

6.112.2.3 [u_int8_t ng_hci_lp_qos_req_ep::flags](#)

Definition at line 449 of file ng_hci.h.

6.112.2.4 [u_int32_t ng_hci_lp_qos_req_ep::latency](#)

Definition at line 453 of file ng_hci.h.

6.112.2.5 [u_int32_t ng_hci_lp_qos_req_ep::peak_bandwidth](#)

Definition at line 452 of file ng_hci.h.

6.112.2.6 [u_int8_t ng_hci_lp_qos_req_ep::service_type](#)

Definition at line 450 of file ng_hci.h.

6.112.2.7 `u_int32_t ng_hci_lp_qos_req_ep::token_rate`

Definition at line 451 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.113 ng_hci_neighbor Struct Reference

```
#include <ng_hci_var.h>
```

Public Member Functions

- [LIST_ENTRY](#) ([ng_hci_neighbor](#)) next

Data Fields

- timeval [updated](#)
- [bdaddr_t](#) [bdaddr](#)
- [u_int8_t](#) [features](#) [NG_HCI_FEATURES_SIZE]
- [u_int8_t](#) [page_scan_rep_mode](#)
- [u_int8_t](#) [page_scan_mode](#)
- [u_int16_t](#) [clock_offset](#)

6.113.1 Detailed Description

Definition at line 202 of file `ng_hci_var.h`.

6.113.2 Member Function Documentation

6.113.2.1 [ng_hci_neighbor::LIST_ENTRY](#) ([ng_hci_neighbor](#))

6.113.3 Field Documentation

6.113.3.1 [bdaddr_t](#) [ng_hci_neighbor::bdaddr](#)

Definition at line 205 of file `ng_hci_var.h`.

Referenced by `inquiry_result()`, `ng_hci_default_rcvmsg()`, `ng_hci_get_neighbor()`, `page_scan_mode_change()`, `page_scan_rep_mode_change()`, `read_clock_offset_compl()`, and `read_remote_features_compl()`.

6.113.3.2 [u_int16_t](#) [ng_hci_neighbor::clock_offset](#)

Definition at line 211 of file `ng_hci_var.h`.

Referenced by `inquiry_result()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, and `read_clock_offset_compl()`.

6.113.3.3 [u_int8_t](#) [ng_hci_neighbor::features](#)[NG_HCI_FEATURES_SIZE]

Definition at line 206 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, and `read_remote_features_compl()`.

6.113.3.4 u_int8_t ng_hci_neighbor::page_scan_mode

Definition at line 210 of file ng_hci_var.h.

Referenced by inquiry_result(), ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), and page_scan_mode_change().

6.113.3.5 u_int8_t ng_hci_neighbor::page_scan_rep_mode

Definition at line 209 of file ng_hci_var.h.

Referenced by inquiry_result(), ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), and page_scan_rep_mode_change().

6.113.3.6 struct timeval ng_hci_neighbor::updated

Definition at line 203 of file ng_hci_var.h.

Referenced by inquiry_result(), ng_hci_neighbor_stale(), page_scan_mode_change(), page_scan_rep_mode_change(), read_clock_offset_compl(), and read_remote_features_compl().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h](#)

6.114 ng_hci_node_buffer_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int8_t cmd_free](#)
- [u_int8_t sco_size](#)
- [u_int16_t sco_pkts](#)
- [u_int16_t sco_free](#)
- [u_int16_t acl_size](#)
- [u_int16_t acl_pkts](#)
- [u_int16_t acl_free](#)

6.114.1 Detailed Description

Definition at line 509 of file ng_hci.h.

6.114.2 Field Documentation

6.114.2.1 [u_int16_t ng_hci_node_buffer_ep::acl_free](#)

Definition at line 516 of file ng_hci.h.

Referenced by [ng_hci_default_rcvmsg\(\)](#).

6.114.2.2 [u_int16_t ng_hci_node_buffer_ep::acl_pkts](#)

Definition at line 515 of file ng_hci.h.

Referenced by [ng_hci_default_rcvmsg\(\)](#).

6.114.2.3 [u_int16_t ng_hci_node_buffer_ep::acl_size](#)

Definition at line 514 of file ng_hci.h.

Referenced by [ng_hci_default_rcvmsg\(\)](#).

6.114.2.4 [u_int8_t ng_hci_node_buffer_ep::cmd_free](#)

Definition at line 510 of file ng_hci.h.

Referenced by [ng_hci_default_rcvmsg\(\)](#).

6.114.2.5 [u_int16_t ng_hci_node_buffer_ep::sco_free](#)

Definition at line 513 of file ng_hci.h.

Referenced by [ng_hci_default_rcvmsg\(\)](#).

6.114.2.6 `u_int16_t ng_hci_node_buffer_ep::sco_pkts`

Definition at line 512 of file `ng_hci.h`.

Referenced by `ng_hci_default_rcvmsg()`.

6.114.2.7 `u_int8_t ng_hci_node_buffer_ep::sco_size`

Definition at line 511 of file `ng_hci.h`.

Referenced by `ng_hci_default_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.115 ng_hci_node_con_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int8_t link_type](#)
- [u_int8_t encryption_mode](#)
- [u_int8_t mode](#)
- [u_int8_t role](#)
- [u_int16_t state](#)
- [u_int16_t reserved](#)
- [u_int16_t pending](#)
- [u_int16_t queue_len](#)
- [u_int16_t con_handle](#)
- [bdaddr_t bdaddr](#)

6.115.1 Detailed Description

Definition at line 565 of file ng_hci.h.

6.115.2 Field Documentation

6.115.2.1 [bdaddr_t ng_hci_node_con_ep::bdaddr](#)

Definition at line 575 of file ng_hci.h.

6.115.2.2 [u_int16_t ng_hci_node_con_ep::con_handle](#)

Definition at line 574 of file ng_hci.h.

6.115.2.3 [u_int8_t ng_hci_node_con_ep::encryption_mode](#)

Definition at line 567 of file ng_hci.h.

6.115.2.4 [u_int8_t ng_hci_node_con_ep::link_type](#)

Definition at line 566 of file ng_hci.h.

6.115.2.5 [u_int8_t ng_hci_node_con_ep::mode](#)

Definition at line 568 of file ng_hci.h.

6.115.2.6 [u_int16_t ng_hci_node_con_ep::pending](#)

Definition at line 572 of file ng_hci.h.

6.115.2.7 `u_int16_t ng_hci_node_con_ep::queue_len`

Definition at line 573 of file `ng_hci.h`.

6.115.2.8 `u_int16_t ng_hci_node_con_ep::reserved`

Definition at line 571 of file `ng_hci.h`.

6.115.2.9 `u_int8_t ng_hci_node_con_ep::role`

Definition at line 569 of file `ng_hci.h`.

6.115.2.10 `u_int16_t ng_hci_node_con_ep::state`

Definition at line 570 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.116 ng_hci_node_con_list_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int32_t num_connections`

6.116.1 Detailed Description

Definition at line 561 of file `ng_hci.h`.

6.116.2 Field Documentation

6.116.2.1 `u_int32_t ng_hci_node_con_list_ep::num_connections`

Definition at line 562 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.117 `ng_hci_node_get_neighbor_cache_ep` Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int32_t num_entries`

6.117.1 Detailed Description

Definition at line 545 of file `ng_hci.h`.

6.117.2 Field Documentation

6.117.2.1 `u_int32_t ng_hci_node_get_neighbor_cache_ep::num_entries`

Definition at line 546 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.118 ng_hci_node_neighbor_cache_entry_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t page_scan_rep_mode](#)
- [u_int16_t page_scan_mode](#)
- [u_int16_t clock_offset](#)
- [bdaddr_t bdaddr](#)
- [u_int8_t features](#) [NG_HCI_FEATURES_SIZE]

6.118.1 Detailed Description

Definition at line 549 of file `ng_hci.h`.

6.118.2 Field Documentation

6.118.2.1 [bdaddr_t ng_hci_node_neighbor_cache_entry_ep::bdaddr](#)

Definition at line 553 of file `ng_hci.h`.

6.118.2.2 [u_int16_t ng_hci_node_neighbor_cache_entry_ep::clock_offset](#)

Definition at line 552 of file `ng_hci.h`.

6.118.2.3 [u_int8_t ng_hci_node_neighbor_cache_entry_ep::features](#)[NG_HCI_FEATURES_SIZE]

Definition at line 554 of file `ng_hci.h`.

6.118.2.4 [u_int16_t ng_hci_node_neighbor_cache_entry_ep::page_scan_mode](#)

Definition at line 551 of file `ng_hci.h`.

6.118.2.5 [u_int16_t ng_hci_node_neighbor_cache_entry_ep::page_scan_rep_mode](#)

Definition at line 550 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.119 ng_hci_node_stat_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int32_t cmd_sent](#)
- [u_int32_t evnt_rcv](#)
- [u_int32_t acl_rcv](#)
- [u_int32_t acl_sent](#)
- [u_int32_t sco_rcv](#)
- [u_int32_t sco_sent](#)
- [u_int32_t bytes_rcv](#)
- [u_int32_t bytes_sent](#)

6.119.1 Detailed Description

Definition at line 528 of file ng_hci.h.

6.119.2 Field Documentation

6.119.2.1 [u_int32_t ng_hci_node_stat_ep::acl_rcv](#)

Definition at line 531 of file ng_hci.h.

6.119.2.2 [u_int32_t ng_hci_node_stat_ep::acl_sent](#)

Definition at line 532 of file ng_hci.h.

6.119.2.3 [u_int32_t ng_hci_node_stat_ep::bytes_rcv](#)

Definition at line 535 of file ng_hci.h.

6.119.2.4 [u_int32_t ng_hci_node_stat_ep::bytes_sent](#)

Definition at line 536 of file ng_hci.h.

6.119.2.5 [u_int32_t ng_hci_node_stat_ep::cmd_sent](#)

Definition at line 529 of file ng_hci.h.

6.119.2.6 [u_int32_t ng_hci_node_stat_ep::evnt_rcv](#)

Definition at line 530 of file ng_hci.h.

6.119.2.7 `u_int32_t ng_hci_node_stat_ep::sco_recv`

Definition at line 533 of file `ng_hci.h`.

6.119.2.8 `u_int32_t ng_hci_node_stat_ep::sco_sent`

Definition at line 534 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.120 ng_hci_node_up_ep Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- [u_int16_t pkt_size](#)
- [u_int16_t num_pkts](#)
- [u_int16_t reserved](#)
- [bdaddr_t bdaddr](#)

6.120.1 Detailed Description

Definition at line 582 of file `ng_hci.h`.

6.120.2 Field Documentation

6.120.2.1 [bdaddr_t ng_hci_node_up_ep::bdaddr](#)

Definition at line 586 of file `ng_hci.h`.

6.120.2.2 [u_int16_t ng_hci_node_up_ep::num_pkts](#)

Definition at line 584 of file `ng_hci.h`.

6.120.2.3 [u_int16_t ng_hci_node_up_ep::pkt_size](#)

Definition at line 583 of file `ng_hci.h`.

6.120.2.4 [u_int16_t ng_hci_node_up_ep::reserved](#)

Definition at line 585 of file `ng_hci.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.121 `ng_hci_sync_con_queue_ep` Struct Reference

```
#include <ng_hci.h>
```

Data Fields

- `u_int16_t con_handle`
- `u_int16_t completed`

6.121.1 Detailed Description

Definition at line 590 of file `ng_hci.h`.

6.121.2 Field Documentation

6.121.2.1 `u_int16_t ng_hci_sync_con_queue_ep::completed`

Definition at line 592 of file `ng_hci.h`.

6.121.2.2 `u_int16_t ng_hci_sync_con_queue_ep::con_handle`

Definition at line 591 of file `ng_hci.h`.

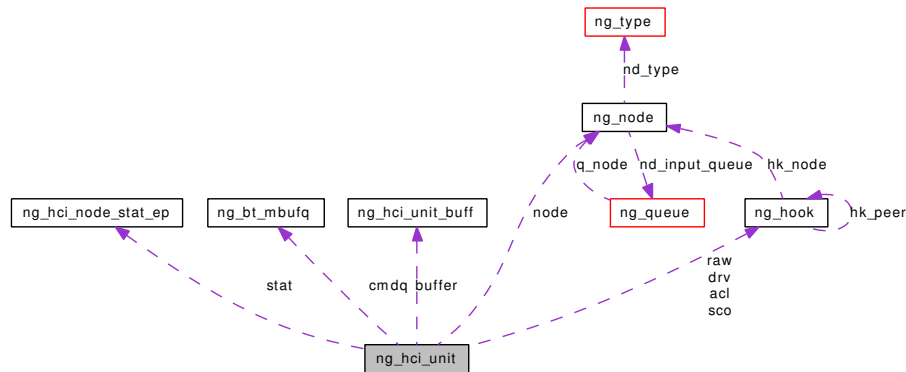
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_hci.h`

6.122 ng_hci_unit Struct Reference

```
#include <ng_hci_var.h>
```

Collaboration diagram for ng_hci_unit:



Public Member Functions

- `LIST_HEAD` (`, ng_hci_unit_con`) `con_list`
- `LIST_HEAD` (`, ng_hci_neighbor`) `neighbors`

Data Fields

- `node_p` `node`
- `ng_hci_node_debug_ep` `debug`
- `ng_hci_node_state_ep` `state`
- `bdaddr_t` `bdaddr`
- `u_int8_t` `features` [`NG_HCI_FEATURES_SIZE`]
- `ng_hci_node_link_policy_mask_ep` `link_policy_mask`
- `ng_hci_node_packet_mask_ep` `packet_mask`
- `ng_hci_node_role_switch_ep` `role_switch`
- `ng_hci_node_stat_ep` `stat`
- `ng_hci_unit_buff_t` `buffer`
- callout `cmd_timo`
- `ng_bt_mbufq_t` `cmdq`
- `hook_p` `drv`
- `hook_p` `acl`
- `hook_p` `sco`
- `hook_p` `raw`

6.122.1 Detailed Description

Definition at line 126 of file `ng_hci_var.h`.

6.122.2 Member Function Documentation

6.122.2.1 `ng_hci_unit::LIST_HEAD (ng_hci_neighbor)`

6.122.2.2 `ng_hci_unit::LIST_HEAD (ng_hci_unit_con)`

6.122.3 Field Documentation

6.122.3.1 `hook_p ng_hci_unit::acl`

Definition at line 158 of file `ng_hci_var.h`.

Referenced by `ng_hci_connect()`, `ng_hci_default_rcvmsg()`, `ng_hci_disconnect()`, `ng_hci_drv_rcvdata()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_ind()`, `ng_hci_lp_qos_req()`, `ng_hci_newhook()`, `ng_hci_node_is_up()`, `process_info_params()`, and `sync_con_queue()`.

6.122.3.2 `bdaddr_t ng_hci_unit::bdaddr`

Definition at line 132 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, `ng_hci_node_is_up()`, `page_scan_mode_change()`, `page_scan_rep_mode_change()`, `process_info_params()`, and `role_change()`.

6.122.3.3 `ng_hci_unit_buff_t ng_hci_unit::buffer`

Definition at line 151 of file `ng_hci_var.h`.

Referenced by `ng_hci_acl_rcvdata()`, `ng_hci_default_rcvmsg()`, `ng_hci_free_con()`, `ng_hci_new_con()`, `ng_hci_node_is_up()`, `ng_hci_process_command_complete()`, `ng_hci_process_command_status()`, `ng_hci_process_command_timeout()`, `ng_hci_sco_rcvdata()`, `ng_hci_send_command()`, `ng_hci_send_data()`, `num_compl_pkts()`, `process_hc_baseband_params()`, and `process_info_params()`.

6.122.3.4 `struct callout ng_hci_unit::cmd_timo`

Definition at line 153 of file `ng_hci_var.h`.

Referenced by `ng_hci_command_timeout()`, and `ng_hci_command_untimeout()`.

6.122.3.5 `ng_bt_mbufq_t ng_hci_unit::cmdq`

Definition at line 154 of file `ng_hci_var.h`.

Referenced by `complete_command()`, `con_compl()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_process_command_timeout()`, `ng_hci_raw_rcvdata()`, `ng_hci_send_command()`, and `ng_hci_shutdown()`.

6.122.3.6 `ng_hci_node_debug_ep ng_hci_unit::debug`

Definition at line 129 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`.

6.122.3.7 hook_p ng_hci_unit::drv

Definition at line 157 of file ng_hci_var.h.

Referenced by ng_hci_connect(), ng_hci_default_rcvmsg(), ng_hci_disconnect(), ng_hci_newhook(), ng_hci_send_command(), and send_data_packets().

6.122.3.8 u_int8_t ng_hci_unit::features[NG_HCI_FEATURES_SIZE]

Definition at line 133 of file ng_hci_var.h.

Referenced by con_compl(), ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_sco_con_req(), and process_info_params().

6.122.3.9 ng_hci_node_link_policy_mask_ep ng_hci_unit::link_policy_mask

Definition at line 136 of file ng_hci_var.h.

Referenced by con_compl(), and ng_hci_default_rcvmsg().

6.122.3.10 node_p ng_hci_unit::node

Definition at line 127 of file ng_hci_var.h.

Referenced by complete_command(), data_buffer_overflow(), discon_compl(), encryption_change(), hardware_error(), mode_change(), ng_hci_acl_rcvdata(), ng_hci_command_timeout(), ng_hci_command_untimeout(), ng_hci_con_timeout(), ng_hci_con_untimeout(), ng_hci_connect(), ng_hci_default_rcvmsg(), ng_hci_drv_rcvdata(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_ind(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_ind(), ng_hci_lp_discon_req(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_ind(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_mtap(), ng_hci_node_is_up(), ng_hci_process_command_complete(), ng_hci_process_command_timeout(), ng_hci_process_event(), ng_hci_raw_rcvdata(), ng_hci_sco_rcvdata(), ng_hci_send_command(), ng_hci_send_data(), ng_hci_shutdown(), num_compl_pkts(), process_info_params(), process_link_policy_params(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), read_remote_features_compl(), role_change(), send_data_packets(), and sync_con_queue().

6.122.3.11 ng_hci_node_packet_mask_ep ng_hci_unit::packet_mask

Definition at line 137 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), and ng_hci_lp_sco_con_req().

6.122.3.12 hook_p ng_hci_unit::raw

Definition at line 160 of file ng_hci_var.h.

Referenced by ng_hci_connect(), ng_hci_default_rcvmsg(), ng_hci_disconnect(), ng_hci_mtap(), and ng_hci_newhook().

6.122.3.13 ng_hci_node_role_switch_ep ng_hci_unit::role_switch

Definition at line 138 of file ng_hci_var.h.

Referenced by `con_compl()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, and `ng_hci_lp_con_rsp()`.

6.122.3.14 [hook_p ng_hci_unit::sco](#)

Definition at line 159 of file `ng_hci_var.h`.

Referenced by `ng_hci_connect()`, `ng_hci_default_rcvmsg()`, `ng_hci_disconnect()`, `ng_hci_drv_rcvdata()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_ind()`, `ng_hci_newhook()`, `ng_hci_node_is_up()`, `process_info_params()`, and `sync_con_queue()`.

6.122.3.15 [ng_hci_node_stat_ep ng_hci_unit::stat](#)

Definition at line 140 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, `ng_hci_drv_rcvdata()`, `ng_hci_send_command()`, `ng_hci_send_data()`, and `send_data_packets()`.

6.122.3.16 [ng_hci_node_state_ep ng_hci_unit::state](#)

Definition at line 130 of file `ng_hci_var.h`.

Referenced by `complete_command()`, `con_compl()`, `ng_hci_command_timeout()`, `ng_hci_command_untimeout()`, `ng_hci_connect()`, `ng_hci_default_rcvmsg()`, `ng_hci_disconnect()`, `ng_hci_drv_rcvdata()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_new_con()`, `ng_hci_node_is_up()`, `ng_hci_process_command_timeout()`, `ng_hci_raw_rcvdata()`, `ng_hci_send_command()`, `ng_hci_unit_clean()`, `process_hc_baseband_params()`, `process_info_params()`, and `send_data_packets()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h](#)

6.123 ng_hci_unit_buff Struct Reference

```
#include <ng_hci_var.h>
```

Data Fields

- [u_int8_t cmd_free](#)
- [u_int8_t sco_size](#)
- [u_int16_t sco_pkts](#)
- [u_int16_t sco_free](#)
- [u_int16_t acl_size](#)
- [u_int16_t acl_pkts](#)
- [u_int16_t acl_free](#)

6.123.1 Detailed Description

Definition at line 64 of file ng_hci_var.h.

6.123.2 Field Documentation

6.123.2.1 [u_int16_t ng_hci_unit_buff::acl_free](#)

Definition at line 73 of file ng_hci_var.h.

6.123.2.2 [u_int16_t ng_hci_unit_buff::acl_pkts](#)

Definition at line 72 of file ng_hci_var.h.

6.123.2.3 [u_int16_t ng_hci_unit_buff::acl_size](#)

Definition at line 71 of file ng_hci_var.h.

6.123.2.4 [u_int8_t ng_hci_unit_buff::cmd_free](#)

Definition at line 65 of file ng_hci_var.h.

6.123.2.5 [u_int16_t ng_hci_unit_buff::sco_free](#)

Definition at line 69 of file ng_hci_var.h.

6.123.2.6 [u_int16_t ng_hci_unit_buff::sco_pkts](#)

Definition at line 68 of file ng_hci_var.h.

6.123.2.7 `u_int8_t ng_hci_unit_buff::sco_size`

Definition at line 67 of file `ng_hci_var.h`.

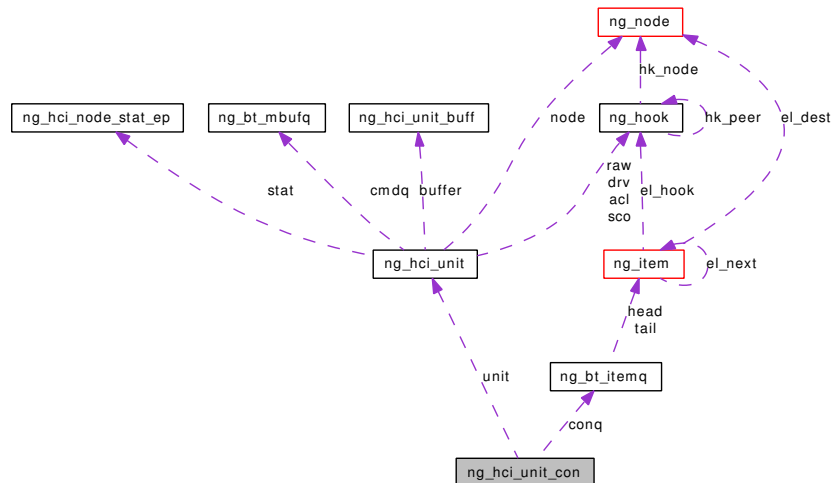
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h](#)

6.124 ng_hci_unit_con Struct Reference

```
#include <ng_hci_var.h>
```

Collaboration diagram for ng_hci_unit_con:



Public Member Functions

- [LIST_ENTRY](#) (ng_hci_unit_con) next

Data Fields

- [ng_hci_unit_p](#) unit
- [u_int16_t](#) state
- [u_int16_t](#) flags
- [bdaddr_t](#) bdaddr
- [u_int16_t](#) con_handle
- [u_int8_t](#) link_type
- [u_int8_t](#) encryption_mode
- [u_int8_t](#) mode
- [u_int8_t](#) role
- callout [con_timo](#)
- int [pending](#)
- [ng_bt_itemq_t](#) conq

6.124.1 Detailed Description

Definition at line 171 of file ng_hci_var.h.

6.124.2 Member Function Documentation

6.124.2.1 `ng_hci_unit_con::LIST_ENTRY` (`ng_hci_unit_con`)

6.124.3 Field Documentation

6.124.3.1 `bdaddr_t ng_hci_unit_con::bdaddr`

Definition at line 180 of file `ng_hci_var.h`.

Referenced by `con_compl()`, `con_req()`, `ng_hci_con_by_bdaddr()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_sco_con_req()`, `read_clock_offset_compl()`, and `read_remote_features_compl()`.

6.124.3.2 `u_int16_t ng_hci_unit_con::con_handle`

Definition at line 181 of file `ng_hci_var.h`.

Referenced by `con_compl()`, `ng_hci_con_by_handle()`, `ng_hci_con_timeout()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_ind()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `num_compl_pkts()`, and `sync_con_queue()`.

6.124.3.3 `struct callout ng_hci_unit_con::con_timo`

Definition at line 188 of file `ng_hci_var.h`.

Referenced by `ng_hci_con_timeout()`, and `ng_hci_con_untimeout()`.

6.124.3.4 `ng_bt_itemq_t ng_hci_unit_con::conq`

Definition at line 191 of file `ng_hci_var.h`.

Referenced by `ng_hci_acl_rcvdata()`, `ng_hci_default_rcvmsg()`, `ng_hci_free_con()`, `ng_hci_sco_rcvdata()`, and `send_data_packets()`.

6.124.3.5 `u_int8_t ng_hci_unit_con::encryption_mode`

Definition at line 184 of file `ng_hci_var.h`.

Referenced by `con_compl()`, `encryption_change()`, and `ng_hci_default_rcvmsg()`.

6.124.3.6 `u_int16_t ng_hci_unit_con::flags`

Definition at line 175 of file `ng_hci_var.h`.

Referenced by `discon_compl()`, `ng_hci_con_timeout()`, `ng_hci_con_untimeout()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_req()`, `ng_hci_process_con_timeout()`, `ng_hci_unit_clean()`, and `process_hc_baseband_params()`.

6.124.3.7 `u_int8_t ng_hci_unit_con::link_type`

Definition at line 183 of file `ng_hci_var.h`.

Referenced by `con_compl()`, `con_req()`, `encryption_change()`, `mode_change()`, `ng_hci_acl_rcvdata()`, `ng_hci_con_by_bdaddr()`, `ng_hci_default_rcvmsg()`, `ng_hci_free_con()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_sco_rcvdata()`, `num_compl_pkts()`, `process_link_policy_params()`, `qos_setup_compl()`, `qos_violation()`, `send_data_packets()`, and `sync_con_queue()`.

6.124.3.8 `u_int8_t ng_hci_unit_con::mode`

Definition at line 185 of file `ng_hci_var.h`.

Referenced by `mode_change()`, and `ng_hci_default_rcvmsg()`.

6.124.3.9 `int ng_hci_unit_con::pending`

Definition at line 190 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, `ng_hci_free_con()`, `num_compl_pkts()`, and `send_data_packets()`.

6.124.3.10 `u_int8_t ng_hci_unit_con::role`

Definition at line 186 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, `process_link_policy_params()`, and `role_change()`.

6.124.3.11 `u_int16_t ng_hci_unit_con::state`

Definition at line 174 of file `ng_hci_var.h`.

Referenced by `con_compl()`, `con_req()`, `ng_hci_acl_rcvdata()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_process_con_timeout()`, `ng_hci_sco_rcvdata()`, `qos_setup_compl()`, and `qos_violation()`.

6.124.3.12 `ng_hci_unit_p ng_hci_unit_con::unit`

Definition at line 172 of file `ng_hci_var.h`.

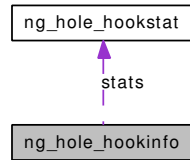
Referenced by `ng_hci_con_timeout()`, `ng_hci_con_untimeout()`, `ng_hci_free_con()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, and `ng_hci_lp_qos_ind()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h`

6.125 ng_hole_hookinfo Struct Reference

Collaboration diagram for ng_hole_hookinfo:



Data Fields

- [ng_hole_hookstat stats](#)

6.125.1 Detailed Description

Definition at line 59 of file ng_hole.c.

6.125.2 Field Documentation

6.125.2.1 struct [ng_hole_hookstat](#) [ng_hole_hookinfo::stats](#)

Definition at line 60 of file ng_hole.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_hole.c](#)

6.126 ng_hole_hookstat Struct Reference

```
#include <ng_hole.h>
```

Data Fields

- [uint64_t frames](#)
- [uint64_t octets](#)

6.126.1 Detailed Description

Definition at line 52 of file `ng_hole.h`.

6.126.2 Field Documentation

6.126.2.1 [uint64_t ng_hole_hookstat::frames](#)

Definition at line 53 of file `ng_hole.h`.

6.126.2.2 [uint64_t ng_hole_hookstat::octets](#)

Definition at line 54 of file `ng_hole.h`.

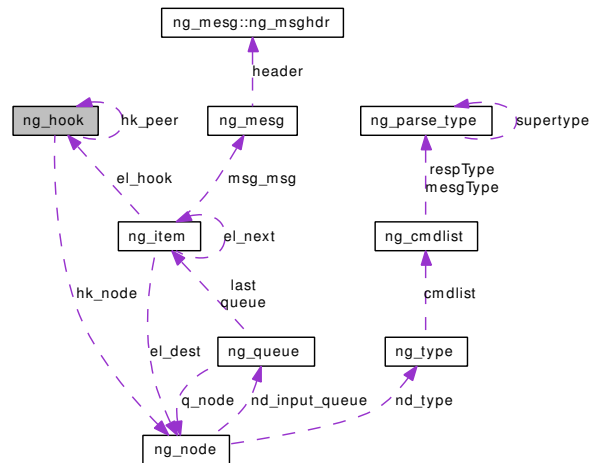
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_hole.h`

6.127 ng_hook Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_hook:



Public Member Functions

- [LIST_ENTRY](#) (ng_hook) hk_hooks

Data Fields

- char [hk_name](#) [NG_HOOKSIZ]
- void * [hk_private](#)
- int [hk_flags](#)
- int [hk_refs](#)
- int [hk_type](#)
- [ng_hook](#) * [hk_peer](#)
- [ng_node](#) * [hk_node](#)
- [ng_rcvmsg_t](#) * [hk_rcvmsg](#)
- [ng_rcvdata_t](#) * [hk_rcvdata](#)

6.127.1 Detailed Description

Definition at line 106 of file netgraph.h.

6.127.2 Member Function Documentation

6.127.2.1 ng_hook::LIST_ENTRY (ng_hook)

6.127.3 Field Documentation

6.127.3.1 int ng_hook::hk_flags

Definition at line 109 of file netgraph.h.

Referenced by ng_add_hook(), ng_con_nodes(), ng_con_part2(), ng_con_part3(), ng_destroy_hook(), ng_mkpeer(), and ng_snd_item().

6.127.3.2 char ng_hook::hk_name[NG_HOOKSIZ]

Definition at line 107 of file netgraph.h.

Referenced by ng_con_part2(), ng_pppoe_rcvdata(), and pppoe_ticker().

6.127.3.3 struct ng_node* ng_hook::hk_node

Definition at line 113 of file netgraph.h.

Referenced by ng_add_hook(), ng_bypass(), ng_con_nodes(), ng_con_part2(), ng_con_part3(), and ng_mkpeer().

6.127.3.4 struct ng_hook* ng_hook::hk_peer

Definition at line 112 of file netgraph.h.

Referenced by ng_add_hook(), ng_bypass(), ng_con_nodes(), ng_con_part2(), ng_destroy_hook(), and ng_mkpeer().

6.127.3.5 void* ng_hook::hk_private

Definition at line 108 of file netgraph.h.

6.127.3.6 ng_rcvdata_t* ng_hook::hk_rcvdata

Definition at line 116 of file netgraph.h.

Referenced by ng_apply_item().

6.127.3.7 ng_rcvmsg_t* ng_hook::hk_rcvmsg

Definition at line 115 of file netgraph.h.

6.127.3.8 int ng_hook::hk_refs

Definition at line 110 of file netgraph.h.

Referenced by ng_add_hook(), ng_con_nodes(), and ng_unref_hook().

6.127.3.9 int [ng_hook::hk_type](#)

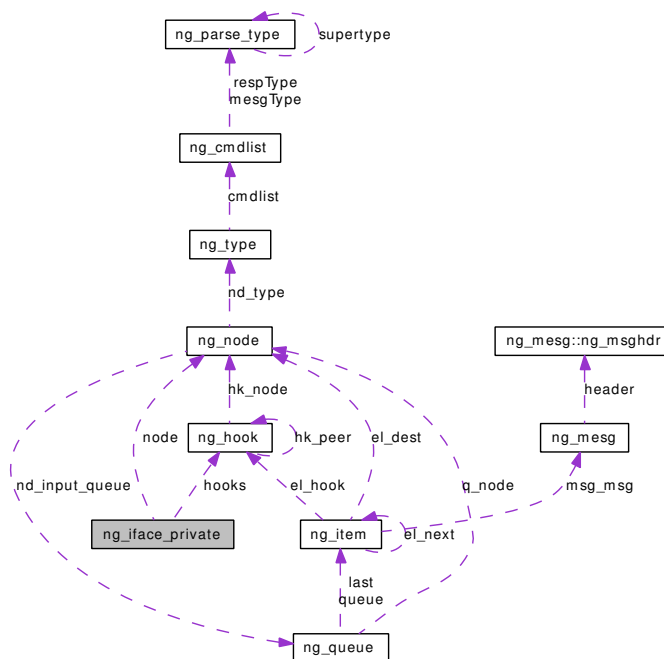
Definition at line 111 of file netgraph.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netgraph.h](#)

6.128 ng_iface_private Struct Reference

Collaboration diagram for ng_iface_private:



Data Fields

- ifnet * ifp
- int unit
- node_p node
- hook_p hooks [NUM_FAMILIES]

6.128.1 Detailed Description

Definition at line 111 of file ng_iface.c.

6.128.2 Field Documentation

6.128.2.1 hook_p ng_iface_private::hooks[[NUM_FAMILIES](#)]

Definition at line 115 of file ng_iface.c.

6.128.2.2 struct ifnet* ng_iface_private::ifp

Definition at line 112 of file ng_iface.c.

6.128.2.3 `node_p ng_iface_private::node`

Definition at line 114 of file `ng_iface.c`.

6.128.2.4 `int ng_iface_private::unit`

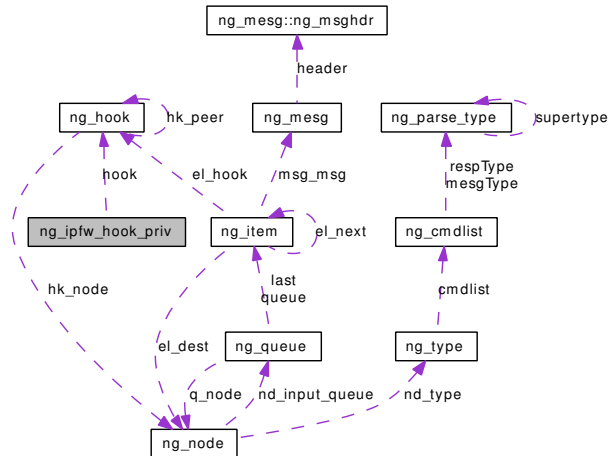
Definition at line 113 of file `ng_iface.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_iface.c](#)

6.129 ng_ipfw_hook_priv Struct Reference

Collaboration diagram for ng_ipfw_hook_priv:



Data Fields

- [hook_p](#) hook
- [u_int16_t](#) rulenum

6.129.1 Detailed Description

Definition at line 86 of file ng_ipfw.c.

6.129.2 Field Documentation

6.129.2.1 [hook_p](#) ng_ipfw_hook_priv::hook

Definition at line 87 of file ng_ipfw.c.

Referenced by ng_ipfw_connect(), ng_ipfw_findhook1(), and ng_ipfw_newhook().

6.129.2.2 [u_int16_t](#) ng_ipfw_hook_priv::rulenum

Definition at line 88 of file ng_ipfw.c.

Referenced by ng_ipfw_newhook().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/ng_ipfw.c

6.130 ng_ipfw_tag Struct Reference

```
#include <ng_ipfw.h>
```

Data Fields

- m_tag [mt](#)
- ip_fw * [rule](#)
- ifnet * [ifp](#)
- int [dir](#)

6.130.1 Detailed Description

Definition at line 38 of file [ng_ipfw.h](#).

6.130.2 Field Documentation

6.130.2.1 int [ng_ipfw_tag::dir](#)

Definition at line 42 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_input\(\)](#), and [ng_ipfw_revdata\(\)](#).

6.130.2.2 struct ifnet* [ng_ipfw_tag::ifp](#)

Definition at line 41 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_input\(\)](#).

6.130.2.3 struct m_tag [ng_ipfw_tag::mt](#)

Definition at line 39 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_input\(\)](#).

6.130.2.4 struct ip_fw* [ng_ipfw_tag::rule](#)

Definition at line 40 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_input\(\)](#).

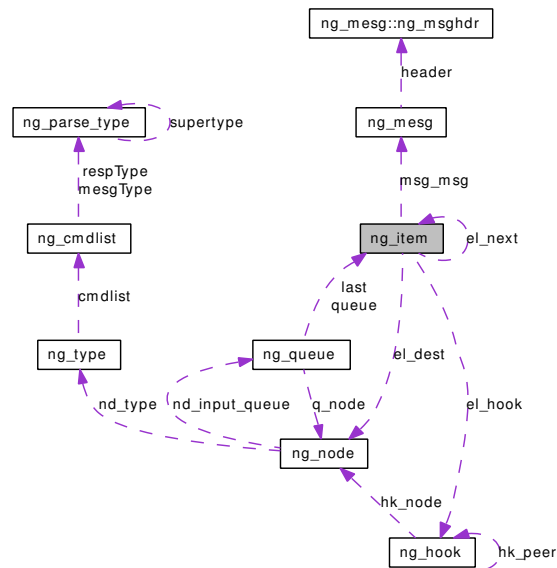
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ipfw.h](#)

6.131 ng_item Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_item:



Data Fields

- `u_long el_flags`
- `item_p el_next`
- `node_p el_dest`
- `hook_p el_hook`
- union {
 - `mbuf * da_m`
 - struct {
 - `ng_mesg * msg_msg`
 - `ng_ID_t msg_retaddr`
 - } `msg`
 - struct {
 - `ng_item_fn * fn_fn`
 - `void * fn_arg1`
 - `int fn_arg2`
 - } `fn`
- } `body`
- `ng_apply_t * apply`
- `void * context`

6.131.1 Detailed Description

Definition at line 583 of file netgraph.h.

6.131.2 Field Documentation

6.131.2.1 `ng_apply_t* ng_item::apply`

Definition at line 604 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_flush_input_queue()`, `ng_free_item()`, and `ngc_send()`.

6.131.2.2 `union { ... } ng_item::body`

6.131.2.3 `void* ng_item::context`

Definition at line 605 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_flush_input_queue()`, and `ngc_send()`.

6.131.2.4 `struct mbuf* ng_item::da_m`

Definition at line 589 of file netgraph.h.

6.131.2.5 `node_p ng_item::el_dest`

Definition at line 586 of file netgraph.h.

Referenced by `ng_connect_data()`, and `ngc_send()`.

6.131.2.6 `u_long ng_item::el_flags`

Definition at line 584 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_raw_input()`, `ng_dequeue()`, `ng_free_item()`, `ng_package_data()`, `ng_package_msg()`, `ng_package_msg_self()`, and `ng_snd_item()`.

6.131.2.7 `hook_p ng_item::el_hook`

Definition at line 587 of file netgraph.h.

6.131.2.8 `item_p ng_item::el_next`

Definition at line 585 of file netgraph.h.

Referenced by `ng_dequeue()`, `ng_flush_input_queue()`, `ng_package_data()`, `ng_package_msg()`, and `ng_package_msg_self()`.

6.131.2.9 `struct { ... } ng_item::fn`

6.131.2.10 `void* ng_item::fn_arg1`

Definition at line 596 of file netgraph.h.

6.131.2.11 `int ng_item::fn_arg2`

Definition at line 597 of file `netgraph.h`.

6.131.2.12 `ng_item_fn* ng_item::fn_fn`

Definition at line 595 of file `netgraph.h`.

6.131.2.13 `struct { ... } ng_item::msg`**6.131.2.14** `struct ng_msg* ng_item::msg_msg`

Definition at line 591 of file `netgraph.h`.

6.131.2.15 `ng_ID_t ng_item::msg_retaddr`

Definition at line 592 of file `netgraph.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netgraph.h`

6.132 ng_ksocket_accept Struct Reference

```
#include <ng_ksocket.h>
```

Data Fields

- `u_int32_t` [nodeid](#)
- `sockaddr` [addr](#)

6.132.1 Detailed Description

Definition at line 74 of file `ng_ksocket.h`.

6.132.2 Field Documentation

6.132.2.1 struct `sockaddr` [ng_ksocket_accept::addr](#)

Definition at line 76 of file `ng_ksocket.h`.

Referenced by `ng_ksocket_finish_accept()`.

6.132.2.2 `u_int32_t` [ng_ksocket_accept::nodeid](#)

Definition at line 75 of file `ng_ksocket.h`.

Referenced by `ng_ksocket_finish_accept()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_ksocket.h`

6.133 ng_ksocket_alias Struct Reference

Data Fields

- const char * [name](#)
- const int [value](#)
- const int [family](#)

6.133.1 Detailed Description

Definition at line 110 of file [ng_ksocket.c](#).

6.133.2 Field Documentation

6.133.2.1 const int [ng_ksocket_alias::family](#)

Definition at line 113 of file [ng_ksocket.c](#).

6.133.2.2 const char* [ng_ksocket_alias::name](#)

Definition at line 111 of file [ng_ksocket.c](#).

Referenced by [ng_ksocket_parse\(\)](#).

6.133.2.3 const int [ng_ksocket_alias::value](#)

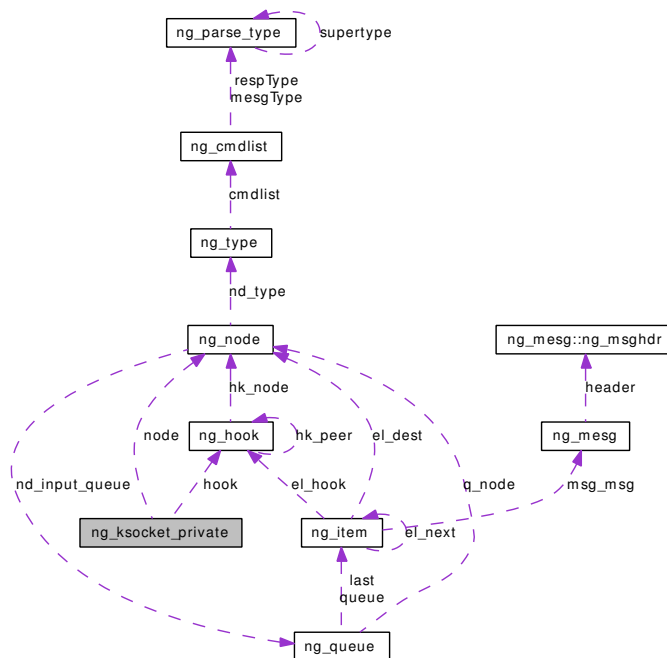
Definition at line 112 of file [ng_ksocket.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ksocket.c](#)

6.134 ng_ksocket_private Struct Reference

Collaboration diagram for ng_ksocket_private:



Data Fields

- `node_p` node
- `hook_p` hook
- socket * `so`

6.134.1 Detailed Description

Definition at line 81 of file `ng_ksocket.c`.

6.134.2 Field Documentation

6.134.2.1 `hook_p ng_ksocket_private::hook`

Definition at line 83 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_connect()`, `ng_ksocket_disconnect()`, `ng_ksocket_newhook()`, and `ng_ksocket_rcvdata()`.

6.134.2.2 `node_p ng_ksocket_private::node`

Definition at line 82 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_connect()`, `ng_ksocket_constructor()`, `ng_ksocket_finish_accept()`, `ng_ksocket_incoming()`, `ng_ksocket_incoming2()`, `ng_ksocket_newhook()`, `ng_ksocket_rcvdata()`, `ng_ksocket_rcvmsg()`, and `ng_ksocket_shutdown()`.

6.134.2.3 `struct socket*` `ng_ksocket_private::so`

Definition at line 84 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_connect()`, `ng_ksocket_finish_accept()`, `ng_ksocket_incoming2()`, `ng_ksocket_rcvdata()`, and `ng_ksocket_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_ksocket.c`

6.135 `ng_ksocket_sockopt` Struct Reference

```
#include <ng_ksocket.h>
```

Data Fields

- `int32_t level`
- `int32_t name`
- `u_char value []`

6.135.1 Detailed Description

Definition at line 54 of file `ng_ksocket.h`.

6.135.2 Field Documentation

6.135.2.1 `int32_t ng_ksocket_sockopt::level`

Definition at line 55 of file `ng_ksocket.h`.

6.135.2.2 `int32_t ng_ksocket_sockopt::name`

Definition at line 56 of file `ng_ksocket.h`.

6.135.2.3 `u_char ng_ksocket_sockopt::value[]`

Definition at line 57 of file `ng_ksocket.h`.

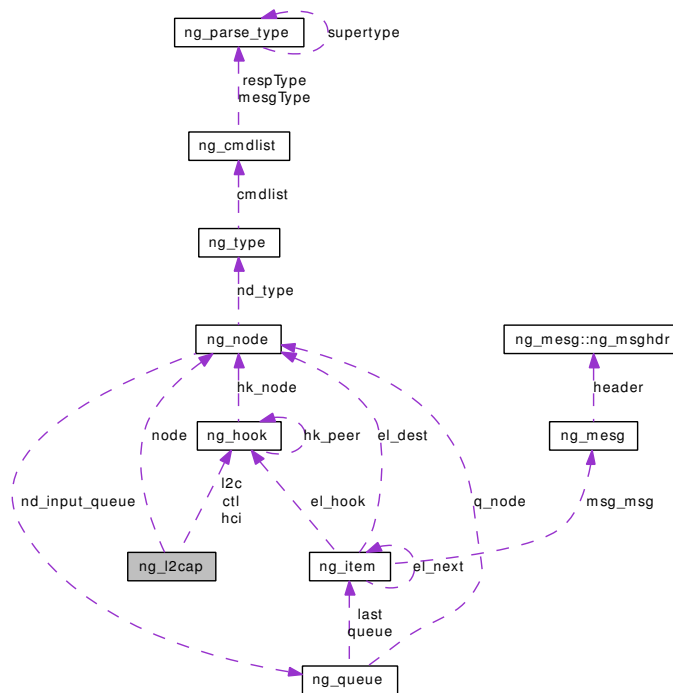
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_ksocket.h`

6.136 ng_l2cap Struct Reference

```
#include <ng_l2cap_var.h>
```

Collaboration diagram for ng_l2cap:



Public Member Functions

- `LIST_HEAD` (`, ng_l2cap_con`) `con_list`
- `LIST_HEAD` (`, ng_l2cap_chan`) `chan_list`

Data Fields

- `node_p` `node`
- `ng_l2cap_node_debug_ep` `debug`
- `ng_l2cap_node_flags_ep` `flags`
- `ng_l2cap_node_auto_discon_ep` `discon_timo`
- `u_int16_t` `pkt_size`
- `u_int16_t` `num_pkts`
- `bdaddr_t` `bdaddr`
- `hook_p` `hci`
- `hook_p` `l2c`
- `hook_p` `ctl`
- `u_int16_t` `cid`

6.136.1 Detailed Description

Definition at line 78 of file `ng_l2cap_var.h`.

6.136.2 Member Function Documentation

6.136.2.1 `ng_l2cap::LIST_HEAD (ng_l2cap_chan)`

6.136.2.2 `ng_l2cap::LIST_HEAD (ng_l2cap_con)`

6.136.3 Field Documentation

6.136.3.1 `bdaddr_t ng_l2cap::bdaddr`

Definition at line 87 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_send_hook_info()`.

6.136.3.2 `u_int16_t ng_l2cap::cid`

Definition at line 95 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_get_cid()`.

6.136.3.3 `hook_p ng_l2cap::ctl`

Definition at line 91 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_connect()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_disconnect()`, `ng_l2cap_l2ca_enable_ctl()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_lower_rcvmsg()`, and `ng_l2cap_newhook()`.

6.136.3.4 `ng_l2cap_node_debug_ep ng_l2cap::debug`

Definition at line 81 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`.

6.136.3.5 `ng_l2cap_node_auto_discon_ep ng_l2cap::discon_timo`

Definition at line 83 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_con_unref()`, `ng_l2cap_default_rcvmsg()`, and `ng_l2cap_discon_timeout()`.

6.136.3.6 `ng_l2cap_node_flags_ep ng_l2cap::flags`

Definition at line 82 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_ctl_receive()`, and `ng_l2cap_l2ca_enable_ctl()`.

6.136.3.7 `hook_p ng_l2cap::hci`

Definition at line 89 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_connect()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_disconnect()`, `ng_l2cap_lp_con_ind()`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `ng_l2cap_lp_qos_req()`, `ng_l2cap_newhook()`, `ng_l2cap_process_discon_timeout()`, `ng_l2cap_revdata()`, and `ng_l2cap_send_hook_info()`.

6.136.3.8 `hook_p ng_l2cap::l2c`

Definition at line 90 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_connect()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_disconnect()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_clt_receive()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_newhook()`, and `ng_l2cap_revdata()`.

6.136.3.9 `node_p ng_l2cap::node`

Definition at line 79 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_cmd_by_ident()`, `ng_l2cap_con_fail()`, `ng_l2cap_con_ref()`, `ng_l2cap_con_unref()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_connect()`, `ng_l2cap_discon_timeout()`, `ng_l2cap_discon_untimeout()`, `ng_l2cap_free_con()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_clt_receive()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `ng_l2cap_lp_discon_ind()`, `ng_l2cap_lp_qos_cfm()`, `ng_l2cap_lp_qos_ind()`, `ng_l2cap_lp_qos_req()`, `ng_l2cap_lp_receive()`, `ng_l2cap_lp_send()`, `ng_l2cap_lp_timeout()`, `ng_l2cap_lp_untimeout()`, `ng_l2cap_new_cmd()`, `ng_l2cap_process_cfg_req()`, `ng_l2cap_process_cfg_rsp()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_req()`, `ng_l2cap_process_discon_rsp()`, `ng_l2cap_process_discon_timeout()`, `ng_l2cap_process_echo_req()`, `ng_l2cap_process_echo_rsp()`, `ng_l2cap_process_info_rsp()`, `ng_l2cap_process_signal_cmd()`, `ng_l2cap_receive()`, `ng_l2cap_send_hook_info()`, and `ng_l2cap_shutdown()`.

6.136.3.10 `u_int16_t ng_l2cap::num_pkts`

Definition at line 86 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

6.136.3.11 `u_int16_t ng_l2cap::pkt_size`

Definition at line 85 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_lower_rcvmsg()`, and `ng_l2cap_lp_send()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h`

6.137 ng_l2cap_cfg_opt_val_t Union Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t mtu](#)
- [u_int16_t flush_timo](#)
- [ng_l2cap_flow_t flow](#)

6.137.1 Detailed Description

Definition at line 248 of file `ng_l2cap.h`.

6.137.2 Field Documentation

6.137.2.1 [ng_l2cap_flow_t ng_l2cap_cfg_opt_val_t::flow](#)

Definition at line 251 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_cfg_req()`, and `ng_l2cap_process_cfg_rsp()`.

6.137.2.2 [u_int16_t ng_l2cap_cfg_opt_val_t::flush_timo](#)

Definition at line 250 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_cfg_req()`, and `ng_l2cap_process_cfg_rsp()`.

6.137.2.3 [u_int16_t ng_l2cap_cfg_opt_val_t::mtu](#)

Definition at line 249 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_cfg_req()`, and `ng_l2cap_process_cfg_rsp()`.

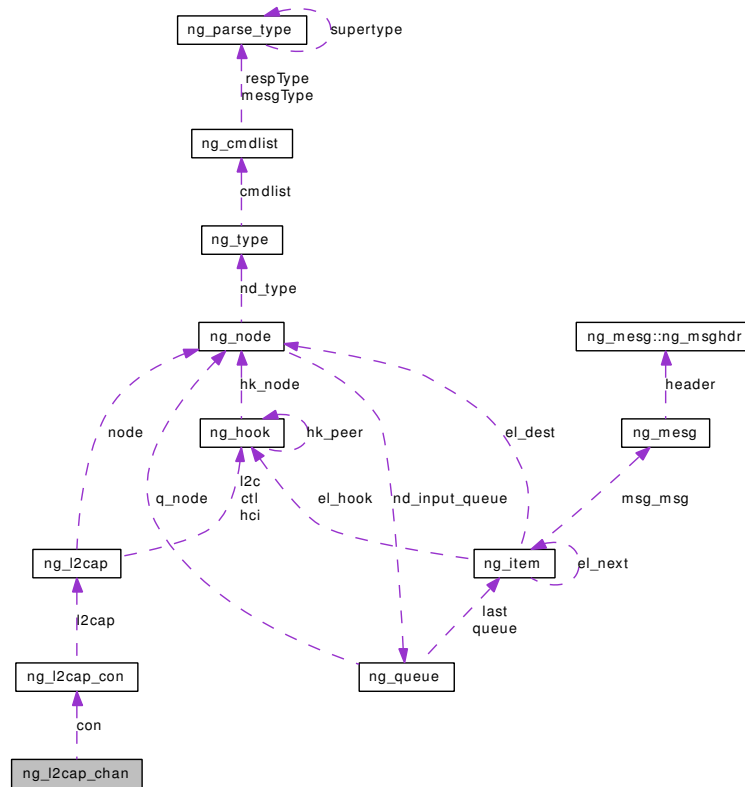
The documentation for this union was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.138 ng_l2cap_chan Struct Reference

```
#include <ng_l2cap_var.h>
```

Collaboration diagram for ng_l2cap_chan:



Public Member Functions

- [LIST_ENTRY \(ng_l2cap_chan\) next](#)

Data Fields

- [ng_l2cap_con_p con](#)
- [u_int16_t state](#)
- [u_int8_t cfg_state](#)
- [u_int8_t ident](#)
- [u_int16_t psm](#)
- [u_int16_t scid](#)
- [u_int16_t dcid](#)
- [u_int16_t imtu](#)
- [ng_l2cap_flow_t iflow](#)
- [u_int16_t omtu](#)
- [ng_l2cap_flow_t oflow](#)
- [u_int16_t flush_timo](#)
- [u_int16_t link_timo](#)

6.138.1 Detailed Description

Definition at line 135 of file `ng_l2cap_var.h`.

6.138.2 Member Function Documentation

6.138.2.1 `ng_l2cap_chan::LIST_ENTRY` ([ng_l2cap_chan](#))

6.138.3 Field Documentation

6.138.3.1 `u_int8_t ng_l2cap_chan::cfg_state`

Definition at line 140 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_rsp_req()`, and `ng_l2cap_process_cfg_req()`.

6.138.3.2 `ng_l2cap_con_p ng_l2cap_chan::con`

Definition at line 136 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_free_chan()`, `ng_l2cap_free_con()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_l2ca_write_rsp()`, and `ng_l2cap_new_cmd()`.

6.138.3.3 `u_int16_t ng_l2cap_chan::dcid`

Definition at line 149 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_process_con_req()`, and `ng_l2cap_process_discon_req()`.

6.138.3.4 `u_int16_t ng_l2cap_chan::flush_timo`

Definition at line 157 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, and `ng_l2cap_process_cfg_req()`.

6.138.3.5 `u_int8_t ng_l2cap_chan::ident`

Definition at line 145 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_new_cmd()`, and `ng_l2cap_process_con_req()`.

6.138.3.6 `ng_l2cap_flow_t ng_l2cap_chan::iflow`

Definition at line 152 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp_req()`, and `ng_l2cap_process_cfg_req()`.

6.138.3.7 `u_int16_t ng_l2cap_chan::imtu`

Definition at line 151 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, and `ng_l2cap_l2ca_receive()`.

6.138.3.8 `u_int16_t ng_l2cap_chan::link_timo`

Definition at line 158 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_req()`.

6.138.3.9 `ng_l2cap_flow_t ng_l2cap_chan::offlow`

Definition at line 155 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, and `ng_l2cap_l2ca_cfg_rsp()`.

6.138.3.10 `u_int16_t ng_l2cap_chan::omtu`

Definition at line 154 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp_req()`, and `ng_l2cap_process_cfg_req()`.

6.138.3.11 `u_int16_t ng_l2cap_chan::psm`

Definition at line 147 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_receive()`, and `ng_l2cap_l2ca_write_rsp()`.

6.138.3.12 `u_int16_t ng_l2cap_chan::scid`

Definition at line 148 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_chan_by_scid()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_process_con_req()`, and `ng_l2cap_process_discon_req()`.

6.138.3.13 `u_int16_t ng_l2cap_chan::state`

Definition at line 138 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_process_cfg_req()`, `ng_l2cap_process_con_req()`, and `ng_l2cap_process_discon_req()`.

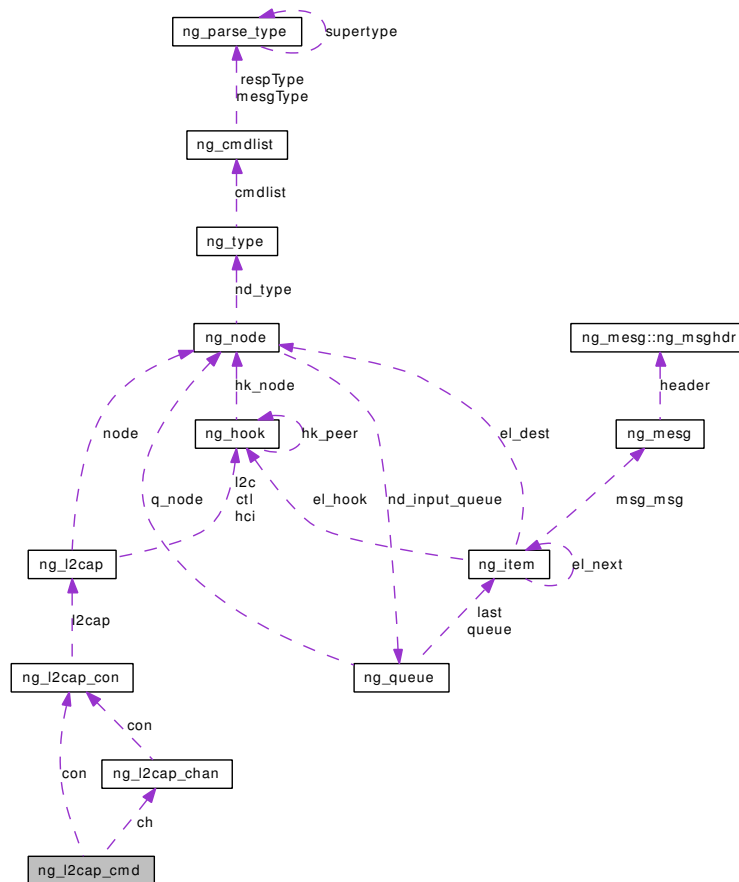
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h](#)

6.139 ng_l2cap_cmd Struct Reference

```
#include <ng_l2cap_var.h>
```

Collaboration diagram for ng_l2cap_cmd:



Public Member Functions

- `TAILQ_ENTRY (ng_l2cap_cmd)` next

Data Fields

- `ng_l2cap_con_p` con
- `ng_l2cap_chan_p` ch
- `u_int16_t` flags
- `u_int8_t` code
- `u_int8_t` ident
- `u_int32_t` token
- callout timo
- `mbuf *` aux

6.139.1 Detailed Description

Definition at line 168 of file `ng_l2cap_var.h`.

6.139.2 Member Function Documentation

6.139.2.1 `ng_l2cap_cmd::TAILQ_ENTRY` (`ng_l2cap_cmd`)

6.139.3 Field Documentation

6.139.3.1 `struct mbuf*` `ng_l2cap_cmd::aux`

Definition at line 181 of file `ng_l2cap_var.h`.

6.139.3.2 `ng_l2cap_chan_p` `ng_l2cap_cmd::ch`

Definition at line 170 of file `ng_l2cap_var.h`.

Referenced by `ng_l2cap_free_chan()`.

6.139.3.3 `u_int8_t` `ng_l2cap_cmd::code`

Definition at line 175 of file `ng_l2cap_var.h`.

6.139.3.4 `ng_l2cap_con_p` `ng_l2cap_cmd::con`

Definition at line 169 of file `ng_l2cap_var.h`.

6.139.3.5 `u_int16_t` `ng_l2cap_cmd::flags`

Definition at line 172 of file `ng_l2cap_var.h`.

6.139.3.6 `u_int8_t` `ng_l2cap_cmd::ident`

Definition at line 176 of file `ng_l2cap_var.h`.

6.139.3.7 `struct callout` `ng_l2cap_cmd::timo`

Definition at line 179 of file `ng_l2cap_var.h`.

6.139.3.8 `u_int32_t` `ng_l2cap_cmd::token`

Definition at line 177 of file `ng_l2cap_var.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h`

6.140 ng_l2cap_cmd_rej_data_t Union Reference

```
#include <ng_l2cap.h>
```

Data Fields

- struct {
 u_int16_t mtu
} mtu
- struct {
 u_int16_t scid
 u_int16_t dcid
} cid

6.140.1 Detailed Description

Definition at line 193 of file ng_l2cap.h.

6.140.2 Field Documentation

6.140.2.1 struct { ... } [ng_l2cap_cmd_rej_data_t::cid](#)

6.140.2.2 u_int16_t [ng_l2cap_cmd_rej_data_t::dcid](#)

Definition at line 201 of file ng_l2cap.h.

6.140.2.3 struct { ... } [ng_l2cap_cmd_rej_data_t::mtu](#)

6.140.2.4 u_int16_t [ng_l2cap_cmd_rej_data_t::mtu](#)

Definition at line 196 of file ng_l2cap.h.

6.140.2.5 u_int16_t [ng_l2cap_cmd_rej_data_t::scid](#)

Definition at line 200 of file ng_l2cap.h.

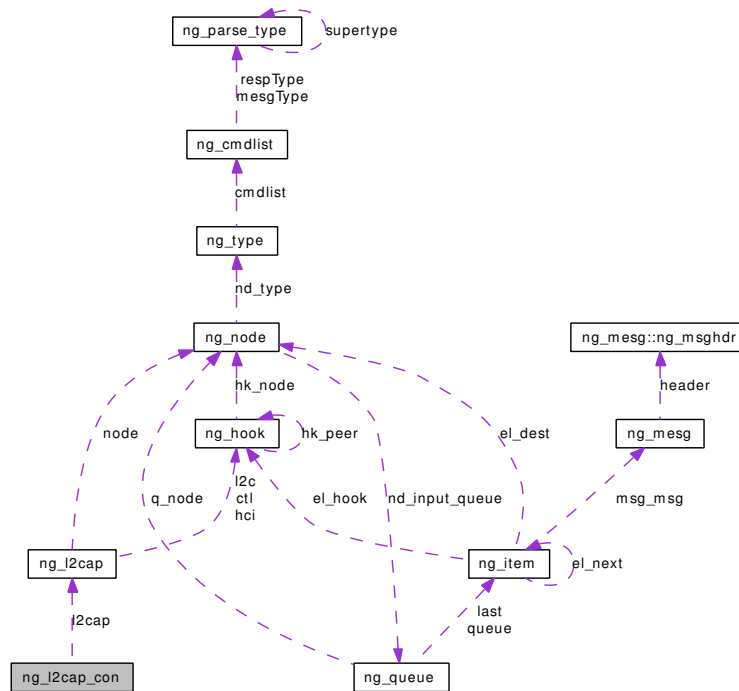
The documentation for this union was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.141 ng_l2cap_con Struct Reference

```
#include <ng_l2cap_var.h>
```

Collaboration diagram for ng_l2cap_con:



Public Member Functions

- [TAILQ_HEAD](#) (`, ng_l2cap_cmd`) `cmd_list`
- [LIST_ENTRY](#) (`ng_l2cap_con`) `next`

Data Fields

- `ng_l2cap_p` `l2cap`
- `u_int16_t` `state`
- `u_int16_t` `flags`
- `int32_t` `refcnt`
- `bdaddr_t` `remote`
- `u_int16_t` `con_handle`
- callout `con_timo`
- `u_int8_t` `ident`
- `mbuf *` `tx_pkt`
- `int` `pending`
- `mbuf *` `rx_pkt`
- `int` `rx_pkt_len`

6.141.1 Detailed Description

Definition at line 106 of file ng_l2cap_var.h.

6.141.2 Member Function Documentation

6.141.2.1 [ng_l2cap_con::LIST_ENTRY \(ng_l2cap_con\)](#)

6.141.2.2 [ng_l2cap_con::TAILQ_HEAD \(ng_l2cap_cmd\)](#)

6.141.3 Field Documentation

6.141.3.1 [u_int16_t ng_l2cap_con::con_handle](#)

Definition at line 115 of file ng_l2cap_var.h.

Referenced by [ng_l2cap_con_by_handle\(\)](#), [ng_l2cap_default_rcvmsg\(\)](#), [ng_l2cap_discon_timeout\(\)](#), [ng_l2cap_lower_rcvmsg\(\)](#), [ng_l2cap_lp_con_cfm\(\)](#), [ng_l2cap_lp_con_ind\(\)](#), [ng_l2cap_lp_con_req\(\)](#), [ng_l2cap_lp_discon_ind\(\)](#), [ng_l2cap_lp_qos_ind\(\)](#), [ng_l2cap_lp_qos_req\(\)](#), [ng_l2cap_lp_receive\(\)](#), [ng_l2cap_lp_send\(\)](#), [ng_l2cap_lp_timeout\(\)](#), [ng_l2cap_process_cfg_rsp\(\)](#), [ng_l2cap_process_con_rsp\(\)](#), [ng_l2cap_process_discon_rsp\(\)](#), and [ng_l2cap_process_discon_timeout\(\)](#).

6.141.3.2 [struct callout ng_l2cap_con::con_timo](#)

Definition at line 116 of file ng_l2cap_var.h.

Referenced by [ng_l2cap_discon_timeout\(\)](#), [ng_l2cap_discon_untimeout\(\)](#), [ng_l2cap_lp_timeout\(\)](#), and [ng_l2cap_lp_untimeout\(\)](#).

6.141.3.3 [u_int16_t ng_l2cap_con::flags](#)

Definition at line 110 of file ng_l2cap_var.h.

Referenced by [ng_l2cap_cleanup\(\)](#), [ng_l2cap_con_fail\(\)](#), [ng_l2cap_con_ref\(\)](#), [ng_l2cap_con_unref\(\)](#), [ng_l2cap_default_rcvmsg\(\)](#), [ng_l2cap_discon_timeout\(\)](#), [ng_l2cap_discon_untimeout\(\)](#), [ng_l2cap_free_con\(\)](#), [ng_l2cap_lp_con_req\(\)](#), [ng_l2cap_lp_discon_ind\(\)](#), [ng_l2cap_lp_timeout\(\)](#), [ng_l2cap_lp_untimeout\(\)](#), [ng_l2cap_process_discon_timeout\(\)](#), and [ng_l2cap_process_lp_timeout\(\)](#).

6.141.3.4 [u_int8_t ng_l2cap_con::ident](#)

Definition at line 118 of file ng_l2cap_var.h.

Referenced by [ng_l2cap_get_ident\(\)](#), and [ng_l2cap_process_signal_cmd\(\)](#).

6.141.3.5 [ng_l2cap_p ng_l2cap_con::l2cap](#)

Definition at line 107 of file ng_l2cap_var.h.

Referenced by [ng_l2cap_cmd_by_ident\(\)](#), [ng_l2cap_con_fail\(\)](#), [ng_l2cap_con_ref\(\)](#), [ng_l2cap_con_unref\(\)](#), [ng_l2cap_con_wakeup\(\)](#), [ng_l2cap_discon_timeout\(\)](#), [ng_l2cap_discon_untimeout\(\)](#), [ng_l2cap_free_con\(\)](#), [ng_l2cap_l2ca_cfg_ind\(\)](#), [ng_l2cap_l2ca_cfg_rsp\(\)](#), [ng_l2cap_l2ca_cfg_rsp_rsp\(\)](#), [ng_l2cap_l2ca_clt_receive\(\)](#), [ng_l2cap_l2ca_con_ind\(\)](#), [ng_l2cap_l2ca_con_rsp\(\)](#), [ng_l2cap_l2ca_con_rsp_rsp\(\)](#),

ng_l2cap_l2ca_discon_ind(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lp_deliver(), ng_l2cap_lp_send(), ng_l2cap_lp_timeout(), ng_l2cap_lp_untimeout(), ng_l2cap_new_cmd(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_echo_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), ng_l2cap_process_signal_cmd(), and ng_l2cap_receive().

6.141.3.6 int ng_l2cap_con::pending

Definition at line 122 of file ng_l2cap_var.h.

Referenced by ng_l2cap_default_rcvmsg(), and ng_l2cap_lower_rcvmsg().

6.141.3.7 int32_t ng_l2cap_con::refcnt

Definition at line 112 of file ng_l2cap_var.h.

Referenced by ng_l2cap_con_ref(), and ng_l2cap_con_unref().

6.141.3.8 bdaddr_t ng_l2cap_con::remote

Definition at line 114 of file ng_l2cap_var.h.

Referenced by ng_l2cap_con_by_addr(), ng_l2cap_default_rcvmsg(), ng_l2cap_l2ca_con_ind(), ng_l2cap_l2ca_ping_rsp(), and ng_l2cap_l2ca_qos_ind().

6.141.3.9 struct mbuf* ng_l2cap_con::rx_pkt

Definition at line 124 of file ng_l2cap_var.h.

Referenced by ng_l2cap_default_rcvmsg(), ng_l2cap_free_con(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_receive(), ng_l2cap_lp_receive(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_echo_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), ng_l2cap_process_signal_cmd(), and ng_l2cap_receive().

6.141.3.10 int ng_l2cap_con::rx_pkt_len

Definition at line 125 of file ng_l2cap_var.h.

Referenced by ng_l2cap_lp_receive().

6.141.3.11 u_int16_t ng_l2cap_con::state

Definition at line 109 of file ng_l2cap_var.h.

Referenced by ng_l2cap_con_ref(), ng_l2cap_con_unref(), ng_l2cap_default_rcvmsg(), ng_l2cap_discon_timeout(), ng_l2cap_discon_untimeout(), ng_l2cap_free_con(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), ng_l2cap_lp_receive(), ng_l2cap_lp_timeout(), ng_l2cap_

lp_untimeout(), ng_l2cap_new_chan(), ng_l2cap_process_discon_timeout(), and ng_l2cap_process_lp_timeout().

6.141.3.12 struct mbuf* [ng_l2cap_con::tx_pkt](#)

Definition at line 121 of file ng_l2cap_var.h.

Referenced by ng_l2cap_default_rcvmsg(), ng_l2cap_free_con(), ng_l2cap_lp_deliver(), and ng_l2cap_lp_send().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h](#)

6.142 ng_l2cap_info_rsp_data_t Union Reference

```
#include <ng_l2cap.h>
```

Data Fields

- struct {
 u_int16_t mtu
} mtu

6.142.1 Detailed Description

Definition at line 293 of file ng_l2cap.h.

6.142.2 Field Documentation

6.142.2.1 struct { ... } [ng_l2cap_info_rsp_data_t::mtu](#)

6.142.2.2 u_int16_t [ng_l2cap_info_rsp_data_t::mtu](#)

Definition at line 296 of file ng_l2cap.h.

The documentation for this union was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.143 ng_l2cap_l2ca_cfg_ind_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)
- [u_int16_t omtu](#)
- [ng_l2cap_flow_t iflow](#)
- [u_int16_t flush_timo](#)

6.143.1 Detailed Description

Definition at line 417 of file `ng_l2cap.h`.

6.143.2 Field Documentation

6.143.2.1 [u_int16_t ng_l2cap_l2ca_cfg_ind_ip::flush_timo](#)

Definition at line 421 of file `ng_l2cap.h`.

6.143.2.2 [ng_l2cap_flow_t ng_l2cap_l2ca_cfg_ind_ip::iflow](#)

Definition at line 420 of file `ng_l2cap.h`.

6.143.2.3 [u_int16_t ng_l2cap_l2ca_cfg_ind_ip::lcid](#)

Definition at line 418 of file `ng_l2cap.h`.

6.143.2.4 [u_int16_t ng_l2cap_l2ca_cfg_ind_ip::omtu](#)

Definition at line 419 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.144 ng_l2cap_l2ca_cfg_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)
- [u_int16_t imtu](#)
- [ng_l2cap_flow_t oflow](#)
- [u_int16_t flush_timo](#)
- [u_int16_t link_timo](#)

6.144.1 Detailed Description

Definition at line 384 of file `ng_l2cap.h`.

6.144.2 Field Documentation

6.144.2.1 [u_int16_t ng_l2cap_l2ca_cfg_ip::flush_timo](#)

Definition at line 388 of file `ng_l2cap.h`.

6.144.2.2 [u_int16_t ng_l2cap_l2ca_cfg_ip::imtu](#)

Definition at line 386 of file `ng_l2cap.h`.

6.144.2.3 [u_int16_t ng_l2cap_l2ca_cfg_ip::lcid](#)

Definition at line 385 of file `ng_l2cap.h`.

6.144.2.4 [u_int16_t ng_l2cap_l2ca_cfg_ip::link_timo](#)

Definition at line 389 of file `ng_l2cap.h`.

6.144.2.5 [ng_l2cap_flow_t ng_l2cap_l2ca_cfg_ip::oflow](#)

Definition at line 387 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.145 ng_l2cap_l2ca_cfg_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)
- [u_int16_t imtu](#)
- [ng_l2cap_flow_t oflow](#)
- [u_int16_t flush_timo](#)

6.145.1 Detailed Description

Definition at line 393 of file ng_l2cap.h.

6.145.2 Field Documentation

6.145.2.1 [u_int16_t ng_l2cap_l2ca_cfg_op::flush_timo](#)

Definition at line 397 of file ng_l2cap.h.

6.145.2.2 [u_int16_t ng_l2cap_l2ca_cfg_op::imtu](#)

Definition at line 395 of file ng_l2cap.h.

6.145.2.3 [ng_l2cap_flow_t ng_l2cap_l2ca_cfg_op::oflow](#)

Definition at line 396 of file ng_l2cap.h.

6.145.2.4 [u_int16_t ng_l2cap_l2ca_cfg_op::result](#)

Definition at line 394 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.146 ng_l2cap_l2ca_cfg_rsp_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)
- [u_int16_t omtu](#)
- [ng_l2cap_flow_t iflow](#)

6.146.1 Detailed Description

Definition at line 403 of file `ng_l2cap.h`.

6.146.2 Field Documentation

6.146.2.1 [ng_l2cap_flow_t ng_l2cap_l2ca_cfg_rsp_ip::iflow](#)

Definition at line 406 of file `ng_l2cap.h`.

6.146.2.2 [u_int16_t ng_l2cap_l2ca_cfg_rsp_ip::lcid](#)

Definition at line 404 of file `ng_l2cap.h`.

6.146.2.3 [u_int16_t ng_l2cap_l2ca_cfg_rsp_ip::omtu](#)

Definition at line 405 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.147 ng_l2cap_l2ca_cfg_rsp_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)

6.147.1 Detailed Description

Definition at line 410 of file ng_l2cap.h.

6.147.2 Field Documentation

6.147.2.1 [u_int16_t ng_l2cap_l2ca_cfg_rsp_op::result](#)

Definition at line 411 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.148 ng_l2cap_l2ca_con_ind_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [bdaddr_t bdaddr](#)
- [u_int16_t lcid](#)
- [u_int16_t psm](#)
- [u_int8_t ident](#)
- [u_int8_t unused](#)

6.148.1 Detailed Description

Definition at line 355 of file ng_l2cap.h.

6.148.2 Field Documentation

6.148.2.1 [bdaddr_t ng_l2cap_l2ca_con_ind_ip::bdaddr](#)

Definition at line 356 of file ng_l2cap.h.

6.148.2.2 [u_int8_t ng_l2cap_l2ca_con_ind_ip::ident](#)

Definition at line 359 of file ng_l2cap.h.

6.148.2.3 [u_int16_t ng_l2cap_l2ca_con_ind_ip::lcid](#)

Definition at line 357 of file ng_l2cap.h.

6.148.2.4 [u_int16_t ng_l2cap_l2ca_con_ind_ip::psm](#)

Definition at line 358 of file ng_l2cap.h.

6.148.2.5 [u_int8_t ng_l2cap_l2ca_con_ind_ip::unused](#)

Definition at line 360 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.149 ng_l2cap_l2ca_con_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t psm](#)
- [bdaddr_t bdaddr](#)

6.149.1 Detailed Description

Definition at line 340 of file ng_l2cap.h.

6.149.2 Field Documentation

6.149.2.1 [bdaddr_t ng_l2cap_l2ca_con_ip::bdaddr](#)

Definition at line 342 of file ng_l2cap.h.

6.149.2.2 [u_int16_t ng_l2cap_l2ca_con_ip::psm](#)

Definition at line 341 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.150 ng_l2cap_l2ca_con_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)
- [u_int16_t result](#)
- [u_int16_t status](#)

6.150.1 Detailed Description

Definition at line 346 of file `ng_l2cap.h`.

6.150.2 Field Documentation

6.150.2.1 [u_int16_t ng_l2cap_l2ca_con_op::lcid](#)

Definition at line 347 of file `ng_l2cap.h`.

6.150.2.2 [u_int16_t ng_l2cap_l2ca_con_op::result](#)

Definition at line 348 of file `ng_l2cap.h`.

6.150.2.3 [u_int16_t ng_l2cap_l2ca_con_op::status](#)

Definition at line 349 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.151 ng_l2cap_l2ca_con_rsp_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [bdaddr_t bdaddr](#)
- [u_int8_t ident](#)
- [u_int8_t unused](#)
- [u_int16_t lcid](#)
- [u_int16_t result](#)
- [u_int16_t status](#)

6.151.1 Detailed Description

Definition at line 367 of file ng_l2cap.h.

6.151.2 Field Documentation

6.151.2.1 [bdaddr_t ng_l2cap_l2ca_con_rsp_ip::bdaddr](#)

Definition at line 368 of file ng_l2cap.h.

6.151.2.2 [u_int8_t ng_l2cap_l2ca_con_rsp_ip::ident](#)

Definition at line 369 of file ng_l2cap.h.

6.151.2.3 [u_int16_t ng_l2cap_l2ca_con_rsp_ip::lcid](#)

Definition at line 371 of file ng_l2cap.h.

6.151.2.4 [u_int16_t ng_l2cap_l2ca_con_rsp_ip::result](#)

Definition at line 372 of file ng_l2cap.h.

6.151.2.5 [u_int16_t ng_l2cap_l2ca_con_rsp_ip::status](#)

Definition at line 373 of file ng_l2cap.h.

6.151.2.6 [u_int8_t ng_l2cap_l2ca_con_rsp_ip::unused](#)

Definition at line 370 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.152 ng_l2cap_l2ca_con_rsp_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)

6.152.1 Detailed Description

Definition at line 377 of file ng_l2cap.h.

6.152.2 Field Documentation

6.152.2.1 u_int16_t ng_l2cap_l2ca_con_rsp_op::result

Definition at line 378 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.153 ng_l2cap_l2ca_discon_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)

6.153.1 Detailed Description

Definition at line 436 of file ng_l2cap.h.

6.153.2 Field Documentation

6.153.2.1 [u_int16_t ng_l2cap_l2ca_discon_ip::lcid](#)

Definition at line 437 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.154 ng_l2cap_l2ca_discon_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)

6.154.1 Detailed Description

Definition at line 441 of file ng_l2cap.h.

6.154.2 Field Documentation

6.154.2.1 [u_int16_t ng_l2cap_l2ca_discon_op::result](#)

Definition at line 442 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.155 ng_l2cap_l2ca_enable_clt_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t psm](#)
- [u_int16_t enable](#)

6.155.1 Detailed Description

Definition at line 560 of file ng_l2cap.h.

6.155.2 Field Documentation

6.155.2.1 [u_int16_t ng_l2cap_l2ca_enable_clt_ip::enable](#)

Definition at line 562 of file ng_l2cap.h.

6.155.2.2 [u_int16_t ng_l2cap_l2ca_enable_clt_ip::psm](#)

Definition at line 561 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.156 ng_l2cap_l2ca_get_info_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [bdaddr_t bdaddr](#)
- [u_int16_t info_type](#)

6.156.1 Detailed Description

Definition at line 545 of file ng_l2cap.h.

6.156.2 Field Documentation

6.156.2.1 [bdaddr_t ng_l2cap_l2ca_get_info_ip::bdaddr](#)

Definition at line 546 of file ng_l2cap.h.

6.156.2.2 [u_int16_t ng_l2cap_l2ca_get_info_ip::info_type](#)

Definition at line 547 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.157 ng_l2cap_l2ca_get_info_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)
- [u_int16_t info_size](#)

6.157.1 Detailed Description

Definition at line 551 of file ng_l2cap.h.

6.157.2 Field Documentation

6.157.2.1 [u_int16_t ng_l2cap_l2ca_get_info_op::info_size](#)

Definition at line 553 of file ng_l2cap.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

6.157.2.2 [u_int16_t ng_l2cap_l2ca_get_info_op::result](#)

Definition at line 552 of file ng_l2cap.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.158 ng_l2cap_l2ca_grp_add_member_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)
- [bdaddr_t bdaddr](#)

6.158.1 Detailed Description

Definition at line 491 of file `ng_l2cap.h`.

6.158.2 Field Documentation

6.158.2.1 [bdaddr_t ng_l2cap_l2ca_grp_add_member_ip::bdaddr](#)

Definition at line 493 of file `ng_l2cap.h`.

6.158.2.2 [u_int16_t ng_l2cap_l2ca_grp_add_member_ip::lcid](#)

Definition at line 492 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.159 ng_l2cap_l2ca_grp_add_member_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)

6.159.1 Detailed Description

Definition at line 497 of file ng_l2cap.h.

6.159.2 Field Documentation

6.159.2.1 u_int16_t ng_l2cap_l2ca_grp_add_member_op::result

Definition at line 498 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.160 ng_l2cap_l2ca_grp_close_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)

6.160.1 Detailed Description

Definition at line 477 of file ng_l2cap.h.

6.160.2 Field Documentation

6.160.2.1 [u_int16_t ng_l2cap_l2ca_grp_close_ip::lcid](#)

Definition at line 478 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.161 ng_l2cap_l2ca_grp_create_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t psm](#)

6.161.1 Detailed Description

Definition at line 465 of file ng_l2cap.h.

6.161.2 Field Documentation

6.161.2.1 [u_int16_t ng_l2cap_l2ca_grp_create_ip::psm](#)

Definition at line 466 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.162 ng_l2cap_l2ca_grp_create_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)

6.162.1 Detailed Description

Definition at line 470 of file ng_l2cap.h.

6.162.2 Field Documentation

6.162.2.1 [u_int16_t ng_l2cap_l2ca_grp_create_op::lcid](#)

Definition at line 471 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.163 ng_l2cap_l2ca_grp_get_members_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t lcid](#)

6.163.1 Detailed Description

Definition at line 514 of file ng_l2cap.h.

6.163.2 Field Documentation

6.163.2.1 [u_int16_t ng_l2cap_l2ca_grp_get_members_ip::lcid](#)

Definition at line 515 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.164 ng_l2cap_l2ca_grp_get_members_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- `u_int16_t` [result](#)
- `u_int16_t` [nmembers](#)

6.164.1 Detailed Description

Definition at line 519 of file `ng_l2cap.h`.

6.164.2 Field Documentation

6.164.2.1 `u_int16_t` [ng_l2cap_l2ca_grp_get_members_op::nmembers](#)

Definition at line 521 of file `ng_l2cap.h`.

6.164.2.2 `u_int16_t` [ng_l2cap_l2ca_grp_get_members_op::result](#)

Definition at line 520 of file `ng_l2cap.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.165 ng_l2cap_l2ca_ping_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [bdaddr_t](#) [bdaddr](#)
- [u_int16_t](#) [echo_size](#)

6.165.1 Detailed Description

Definition at line 528 of file [ng_l2cap.h](#).

6.165.2 Field Documentation

6.165.2.1 [bdaddr_t](#) [ng_l2cap_l2ca_ping_ip::bdaddr](#)

Definition at line 529 of file [ng_l2cap.h](#).

6.165.2.2 [u_int16_t](#) [ng_l2cap_l2ca_ping_ip::echo_size](#)

Definition at line 530 of file [ng_l2cap.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.166 ng_l2cap_l2ca_ping_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int16_t result](#)
- [bdaddr_t bdaddr](#)
- [u_int16_t echo_size](#)

6.166.1 Detailed Description

Definition at line 535 of file ng_l2cap.h.

6.166.2 Field Documentation

6.166.2.1 [bdaddr_t ng_l2cap_l2ca_ping_op::bdaddr](#)

Definition at line 537 of file ng_l2cap.h.

6.166.2.2 [u_int16_t ng_l2cap_l2ca_ping_op::echo_size](#)

Definition at line 538 of file ng_l2cap.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

6.166.2.3 [u_int16_t ng_l2cap_l2ca_ping_op::result](#)

Definition at line 536 of file ng_l2cap.h.

Referenced by [ng_btsocket_l2cap_raw_control\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.167 ng_l2cap_l2ca_qos_ind_ip Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [bdaddr_t bdaddr](#)

6.167.1 Detailed Description

Definition at line 428 of file ng_l2cap.h.

6.167.2 Field Documentation

6.167.2.1 [bdaddr_t ng_l2cap_l2ca_qos_ind_ip::bdaddr](#)

Definition at line 429 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.168 ng_l2cap_l2ca_write_op Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- int [result](#)
- u_int16_t [length](#)
- u_int16_t [lcid](#)

6.168.1 Detailed Description

Definition at line 456 of file ng_l2cap.h.

6.168.2 Field Documentation

6.168.2.1 u_int16_t [ng_l2cap_l2ca_write_op::lcid](#)

Definition at line 459 of file ng_l2cap.h.

6.168.2.2 u_int16_t [ng_l2cap_l2ca_write_op::length](#)

Definition at line 458 of file ng_l2cap.h.

6.168.2.3 int [ng_l2cap_l2ca_write_op::result](#)

Definition at line 457 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.169 ng_l2cap_node_chan_ep Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int32_t state](#)
- [u_int16_t scid](#)
- [u_int16_t dcid](#)
- [u_int16_t imtu](#)
- [u_int16_t omtu](#)
- [u_int16_t psm](#)
- [bdaddr_t remote](#)

6.169.1 Detailed Description

Definition at line 644 of file ng_l2cap.h.

6.169.2 Field Documentation

6.169.2.1 [u_int16_t ng_l2cap_node_chan_ep::dcid](#)

Definition at line 648 of file ng_l2cap.h.

Referenced by [ng_l2cap_default_rcvmsg\(\)](#).

6.169.2.2 [u_int16_t ng_l2cap_node_chan_ep::imtu](#)

Definition at line 650 of file ng_l2cap.h.

Referenced by [ng_l2cap_default_rcvmsg\(\)](#).

6.169.2.3 [u_int16_t ng_l2cap_node_chan_ep::omtu](#)

Definition at line 651 of file ng_l2cap.h.

Referenced by [ng_l2cap_default_rcvmsg\(\)](#).

6.169.2.4 [u_int16_t ng_l2cap_node_chan_ep::psm](#)

Definition at line 653 of file ng_l2cap.h.

Referenced by [ng_l2cap_default_rcvmsg\(\)](#).

6.169.2.5 [bdaddr_t ng_l2cap_node_chan_ep::remote](#)

Definition at line 654 of file ng_l2cap.h.

Referenced by [ng_l2cap_default_rcvmsg\(\)](#).

6.169.2.6 `u_int16_t ng_l2cap_node_chan_ep::scid`

Definition at line 647 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_default_rcvmsg()`.

6.169.2.7 `u_int32_t ng_l2cap_node_chan_ep::state`

Definition at line 645 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_default_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.170 `ng_l2cap_node_chan_list_ep` Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- `u_int32_t num_channels`

6.170.1 Detailed Description

Definition at line 640 of file `ng_l2cap.h`.

6.170.2 Field Documentation

6.170.2.1 `u_int32_t ng_l2cap_node_chan_list_ep::num_channels`

Definition at line 641 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`, and `ng_l2cap_default_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.171 ng_l2cap_node_con_ep Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- [u_int8_t state](#)
- [u_int8_t flags](#)
- [int16_t pending](#)
- [u_int16_t con_handle](#)
- [bdaddr_t remote](#)

6.171.1 Detailed Description

Definition at line 628 of file ng_l2cap.h.

6.171.2 Field Documentation

6.171.2.1 [u_int16_t ng_l2cap_node_con_ep::con_handle](#)

Definition at line 632 of file ng_l2cap.h.

6.171.2.2 [u_int8_t ng_l2cap_node_con_ep::flags](#)

Definition at line 630 of file ng_l2cap.h.

6.171.2.3 [int16_t ng_l2cap_node_con_ep::pending](#)

Definition at line 631 of file ng_l2cap.h.

6.171.2.4 [bdaddr_t ng_l2cap_node_con_ep::remote](#)

Definition at line 633 of file ng_l2cap.h.

6.171.2.5 [u_int8_t ng_l2cap_node_con_ep::state](#)

Definition at line 629 of file ng_l2cap.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h](#)

6.172 `ng_l2cap_node_con_list_ep` Struct Reference

```
#include <ng_l2cap.h>
```

Data Fields

- `u_int32_t num_connections`

6.172.1 Detailed Description

Definition at line 616 of file `ng_l2cap.h`.

6.172.2 Field Documentation

6.172.2.1 `u_int32_t ng_l2cap_node_con_list_ep::num_connections`

Definition at line 617 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h`

6.173 ng_l2tp_config Struct Reference

```
#include <ng_l2tp.h>
```

Data Fields

- u_char [enabled](#)
- u_char [match_id](#)
- u_int16_t [tunnel_id](#)
- u_int16_t [peer_id](#)
- u_int16_t [peer_win](#)
- u_int16_t [rexmit_max](#)
- u_int16_t [rexmit_max_to](#)

6.173.1 Detailed Description

Definition at line 73 of file ng_l2tp.h.

6.173.2 Field Documentation

6.173.2.1 u_char [ng_l2tp_config::enabled](#)

Definition at line 74 of file ng_l2tp.h.

Referenced by [ng_l2tp_seq_adjust\(\)](#).

6.173.2.2 u_char [ng_l2tp_config::match_id](#)

Definition at line 75 of file ng_l2tp.h.

6.173.2.3 u_int16_t [ng_l2tp_config::peer_id](#)

Definition at line 77 of file ng_l2tp.h.

6.173.2.4 u_int16_t [ng_l2tp_config::peer_win](#)

Definition at line 78 of file ng_l2tp.h.

Referenced by [ng_l2tp_seq_adjust\(\)](#).

6.173.2.5 u_int16_t [ng_l2tp_config::rexmit_max](#)

Definition at line 79 of file ng_l2tp.h.

Referenced by [ng_l2tp_seq_adjust\(\)](#).

6.173.2.6 `u_int16_t ng_l2tp_config::remit_max_to`

Definition at line 80 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_seq_adjust()`.

6.173.2.7 `u_int16_t ng_l2tp_config::tunnel_id`

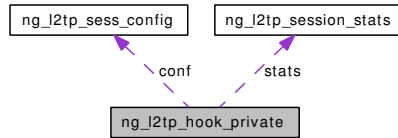
Definition at line 76 of file `ng_l2tp.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_l2tp.h`

6.174 ng_l2tp_hook_private Struct Reference

Collaboration diagram for ng_l2tp_hook_private:



Data Fields

- [ng_l2tp_sess_config](#) `conf`
- [ng_l2tp_session_stats](#) `stats`
- `u_int16_t ns`
- `u_int16_t nr`

6.174.1 Detailed Description

Definition at line 148 of file `ng_l2tp.c`.

6.174.2 Field Documentation

6.174.2.1 struct [ng_l2tp_sess_config](#) `ng_l2tp_hook_private::conf`

Definition at line 149 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_find_session()`, `ng_l2tp_rcvmsg()`, `ng_l2tp_rcv_data()`, `ng_l2tp_rcv_lower()`, and `ng_l2tp_reset_session()`.

6.174.2.2 `u_int16_t` [ng_l2tp_hook_private::nr](#)

Definition at line 152 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_data()`, `ng_l2tp_rcv_lower()`, and `ng_l2tp_reset_session()`.

6.174.2.3 `u_int16_t` [ng_l2tp_hook_private::ns](#)

Definition at line 151 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_data()`, and `ng_l2tp_reset_session()`.

6.174.2.4 struct [ng_l2tp_session_stats](#) `ng_l2tp_hook_private::stats`

Definition at line 150 of file `ng_l2tp.c`.

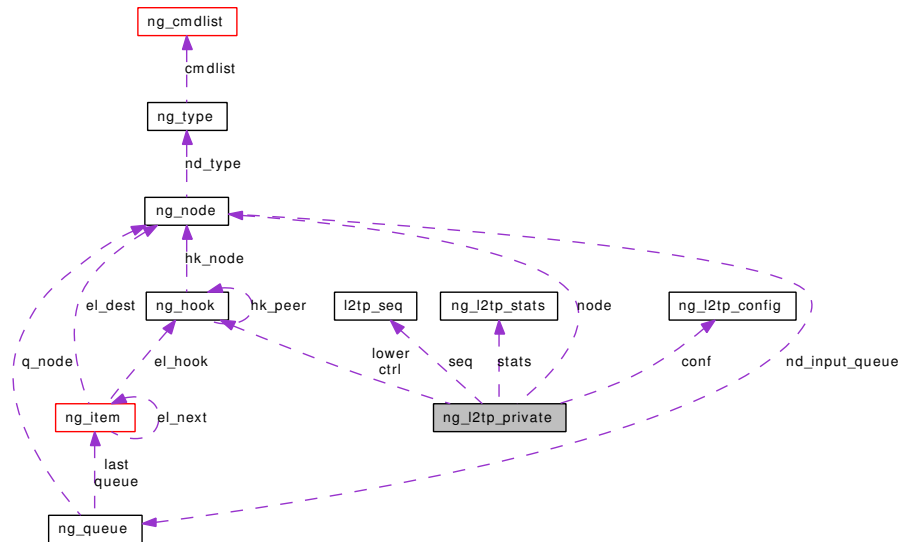
Referenced by `ng_l2tp_rcvmsg()`, `ng_l2tp_rcv_data()`, and `ng_l2tp_rcv_lower()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_l2tp.c`

6.175 ng_l2tp_private Struct Reference

Collaboration diagram for ng_l2tp_private:



Data Fields

- `node_p` `node`
- `hook_p` `ctrl`
- `hook_p` `lower`
- `ng_l2tp_config` `conf`
- `ng_l2tp_stats` `stats`
- `l2tp_seq` `seq`
- `ng_ID_t` `ftarget`

6.175.1 Detailed Description

Definition at line 136 of file `ng_l2tp.c`.

6.175.2 Field Documentation

6.175.2.1 struct `ng_l2tp_config` `ng_l2tp_private::conf`

Definition at line 140 of file `ng_l2tp.c`.

6.175.2.2 `hook_p` `ng_l2tp_private::ctrl`

Definition at line 138 of file `ng_l2tp.c`.

6.175.2.3 `ng_ID_t` `ng_l2tp_private::ftarget`

Definition at line 143 of file `ng_l2tp.c`.

6.175.2.4 [hook_p ng_l2tp_private::lower](#)

Definition at line 139 of file ng_l2tp.c.

6.175.2.5 [node_p ng_l2tp_private::node](#)

Definition at line 137 of file ng_l2tp.c.

6.175.2.6 [struct l2tp_seq ng_l2tp_private::seq](#)

Definition at line 142 of file ng_l2tp.c.

6.175.2.7 [struct ng_l2tp_stats ng_l2tp_private::stats](#)

Definition at line 141 of file ng_l2tp.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_l2tp.c](#)

6.176 ng_l2tp_seq_config Struct Reference

```
#include <ng_l2tp.h>
```

Data Fields

- [u_int16_t ns](#)
- [u_int16_t nr](#)
- [u_int16_t rack](#)
- [u_int16_t xack](#)

6.176.1 Detailed Description

Definition at line 58 of file `ng_l2tp.h`.

6.176.2 Field Documentation

6.176.2.1 [u_int16_t ng_l2tp_seq_config::nr](#)

Definition at line 60 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_seq_set()`.

6.176.2.2 [u_int16_t ng_l2tp_seq_config::ns](#)

Definition at line 59 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_seq_set()`.

6.176.2.3 [u_int16_t ng_l2tp_seq_config::rack](#)

Definition at line 61 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_seq_set()`.

6.176.2.4 [u_int16_t ng_l2tp_seq_config::xack](#)

Definition at line 62 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_seq_set()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_l2tp.h`

6.177 ng_l2tp_sess_config Struct Reference

```
#include <ng_l2tp.h>
```

Data Fields

- `u_int16_t session_id`
- `u_int16_t peer_id`
- `u_char control_dseq`
- `u_char enable_dseq`
- `u_char include_length`

6.177.1 Detailed Description

Definition at line 96 of file `ng_l2tp.h`.

6.177.2 Field Documentation

6.177.2.1 `u_char ng_l2tp_sess_config::control_dseq`

Definition at line 99 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_rcv_lower()`, and `ng_l2tp_reset_session()`.

6.177.2.2 `u_char ng_l2tp_sess_config::enable_dseq`

Definition at line 100 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_rcv_data()`, `ng_l2tp_rcv_lower()`, and `ng_l2tp_reset_session()`.

6.177.2.3 `u_char ng_l2tp_sess_config::include_length`

Definition at line 101 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_rcv_data()`.

6.177.2.4 `u_int16_t ng_l2tp_sess_config::peer_id`

Definition at line 98 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_rcvmsg()`, and `ng_l2tp_rcv_data()`.

6.177.2.5 `u_int16_t ng_l2tp_sess_config::session_id`

Definition at line 97 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_find_session()`, and `ng_l2tp_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_l2tp.h`

6.178 ng_l2tp_session_stats Struct Reference

```
#include <ng_l2tp.h>
```

Data Fields

- [u_int64_t xmitPackets](#)
- [u_int64_t xmitOctets](#)
- [u_int64_t recvPackets](#)
- [u_int64_t recvOctets](#)

6.178.1 Detailed Description

Definition at line 164 of file ng_l2tp.h.

6.178.2 Field Documentation

6.178.2.1 [u_int64_t ng_l2tp_session_stats::recvOctets](#)

Definition at line 168 of file ng_l2tp.h.

Referenced by [ng_l2tp_recv_lower\(\)](#).

6.178.2.2 [u_int64_t ng_l2tp_session_stats::recvPackets](#)

Definition at line 167 of file ng_l2tp.h.

Referenced by [ng_l2tp_recv_lower\(\)](#).

6.178.2.3 [u_int64_t ng_l2tp_session_stats::xmitOctets](#)

Definition at line 166 of file ng_l2tp.h.

Referenced by [ng_l2tp_recv_data\(\)](#).

6.178.2.4 [u_int64_t ng_l2tp_session_stats::xmitPackets](#)

Definition at line 165 of file ng_l2tp.h.

Referenced by [ng_l2tp_recv_data\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_l2tp.h](#)

6.179 ng_l2tp_stats Struct Reference

```
#include <ng_l2tp.h>
```

Data Fields

- [u_int32_t xmitPackets](#)
- [u_int32_t xmitOctets](#)
- [u_int32_t xmitZLBs](#)
- [u_int32_t xmitDrops](#)
- [u_int32_t xmitTooBig](#)
- [u_int32_t xmitInvalid](#)
- [u_int32_t xmitDataTooBig](#)
- [u_int32_t xmitRetransmits](#)
- [u_int32_t recvPackets](#)
- [u_int32_t recvOctets](#)
- [u_int32_t recvRunts](#)
- [u_int32_t recvInvalid](#)
- [u_int32_t recvWrongTunnel](#)
- [u_int32_t recvUnknownSID](#)
- [u_int32_t recvBadAcks](#)
- [u_int32_t recvOutOfOrder](#)
- [u_int32_t recvDuplicates](#)
- [u_int32_t recvDataDrops](#)
- [u_int32_t recvZLBs](#)
- [u_int32_t memoryFailures](#)

6.179.1 Detailed Description

Definition at line 115 of file `ng_l2tp.h`.

6.179.2 Field Documentation

6.179.2.1 [u_int32_t ng_l2tp_stats::memoryFailures](#)

Definition at line 135 of file `ng_l2tp.h`.

6.179.2.2 [u_int32_t ng_l2tp_stats::recvBadAcks](#)

Definition at line 130 of file `ng_l2tp.h`.

6.179.2.3 [u_int32_t ng_l2tp_stats::recvDataDrops](#)

Definition at line 133 of file `ng_l2tp.h`.

6.179.2.4 [u_int32_t ng_l2tp_stats::recvDuplicates](#)

Definition at line 132 of file `ng_l2tp.h`.

6.179.2.5 `u_int32_t ng_l2tp_stats::recvInvalid`

Definition at line 127 of file ng_l2tp.h.

6.179.2.6 `u_int32_t ng_l2tp_stats::recvOctets`

Definition at line 125 of file ng_l2tp.h.

6.179.2.7 `u_int32_t ng_l2tp_stats::recvOutOfOrder`

Definition at line 131 of file ng_l2tp.h.

6.179.2.8 `u_int32_t ng_l2tp_stats::recvPackets`

Definition at line 124 of file ng_l2tp.h.

6.179.2.9 `u_int32_t ng_l2tp_stats::recvRunts`

Definition at line 126 of file ng_l2tp.h.

6.179.2.10 `u_int32_t ng_l2tp_stats::recvUnknownSID`

Definition at line 129 of file ng_l2tp.h.

6.179.2.11 `u_int32_t ng_l2tp_stats::recvWrongTunnel`

Definition at line 128 of file ng_l2tp.h.

6.179.2.12 `u_int32_t ng_l2tp_stats::recvZLBS`

Definition at line 134 of file ng_l2tp.h.

6.179.2.13 `u_int32_t ng_l2tp_stats::xmitDataTooBig`

Definition at line 122 of file ng_l2tp.h.

6.179.2.14 `u_int32_t ng_l2tp_stats::xmitDrops`

Definition at line 119 of file ng_l2tp.h.

6.179.2.15 `u_int32_t ng_l2tp_stats::xmitInvalid`

Definition at line 121 of file ng_l2tp.h.

6.179.2.16 `u_int32_t ng_l2tp_stats::xmitOctets`

Definition at line 117 of file `ng_l2tp.h`.

6.179.2.17 `u_int32_t ng_l2tp_stats::xmitPackets`

Definition at line 116 of file `ng_l2tp.h`.

6.179.2.18 `u_int32_t ng_l2tp_stats::xmitRetransmits`

Definition at line 123 of file `ng_l2tp.h`.

6.179.2.19 `u_int32_t ng_l2tp_stats::xmitTooBig`

Definition at line 120 of file `ng_l2tp.h`.

6.179.2.20 `u_int32_t ng_l2tp_stats::xmitZLBS`

Definition at line 118 of file `ng_l2tp.h`.

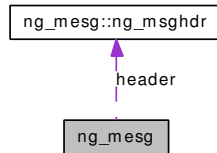
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_l2tp.h`

6.180 ng_mesg Struct Reference

```
#include <ng_message.h>
```

Collaboration diagram for ng_mesg:



Data Fields

- [ng_mesg::ng_msghdr header](#)
- [char data \[\]](#)

Data Structures

- [struct ng_msghdr](#)

6.180.1 Detailed Description

Definition at line 66 of file ng_message.h.

6.180.2 Field Documentation

6.180.2.1 [char ng_mesg::data\[\]](#)

Definition at line 78 of file ng_message.h.

Referenced by [cisco_rcvmsg\(\)](#), [flow_upper\(\)](#), [ng_atm_event_func\(\)](#), [ng_atm_rcvmsg\(\)](#), [ng_atmpif_rcvmsg\(\)](#), [ng_bpf_rcvmsg\(\)](#), [ng_bridge_rcvmsg\(\)](#), [ng_bt3c_rcvmsg\(\)](#), [ng_btsocket_hci_raw_control\(\)](#), [ng_btsocket_hci_raw_send_ngmsg\(\)](#), [ng_btsocket_hci_raw_send_sync_ngmsg\(\)](#), [ng_btsocket_l2cap_default_msg_input\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_req_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_req_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_rsp_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_discon_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_discon_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_write_rsp\(\)](#), [ng_btsocket_l2cap_raw_control\(\)](#), [ng_btsocket_l2cap_raw_input\(\)](#), [ng_btsocket_l2cap_raw_send_ngmsg\(\)](#), [ng_btsocket_l2cap_raw_send_sync_ngmsg\(\)](#), [ng_btsocket_l2cap_send_l2ca_cfg_req\(\)](#), [ng_btsocket_l2cap_send_l2ca_cfg_rsp\(\)](#), [ng_btsocket_l2cap_send_l2ca_con_req\(\)](#), [ng_btsocket_l2cap_send_l2ca_con_rsp_req\(\)](#), [ng_btsocket_l2cap_send_l2ca_discon_req\(\)](#), [ng_ccatm_rcvmsg\(\)](#), [ng_deflate_rcvmsg\(\)](#), [ng_eiface_rcvmsg\(\)](#), [ng_etf_rcvmsg\(\)](#), [ng_ether_rcvmsg\(\)](#), [ng_fec_rcvmsg\(\)](#), [ng_generic_msg\(\)](#), [ng_gif_rcvmsg\(\)](#), [ng_h4_rcvmsg\(\)](#), [ng_hci_default_rcvmsg\(\)](#), [ng_hci_lp_acl_con_req\(\)](#), [ng_hci_lp_con_cfm\(\)](#), [ng_hci_lp_discon_ind\(\)](#), [ng_hci_lp_qos_cfm\(\)](#), [ng_hci_lp_qos_ind\(\)](#), [ng_hci_node_is_up\(\)](#), [ng_iface_rcvmsg\(\)](#), [ng_ksocket_finish_accept\(\)](#), [ng_ksocket_rcvmsg\(\)](#), [ng_l2cap_default_rcvmsg\(\)](#), [ng_l2cap_l2ca_cfg_ind\(\)](#), [ng_l2cap_l2ca_cfg_req\(\)](#), [ng_l2cap_l2ca_cfg_rsp\(\)](#), [ng_l2cap_l2ca_cfg_rsp_req\(\)](#), [ng_l2cap_l2ca_cfg_rsp_rsp\(\)](#), [ng_l2cap_l2ca_con_ind\(\)](#), [ng_l2cap_l2ca_con_req\(\)](#), [ng_l2cap_l2ca_con_rsp\(\)](#), [ng_l2cap_l2ca_con_rsp_req\(\)](#), [ng_l2cap_l2ca_con_rsp_rsp\(\)](#), [ng_l2cap_l2ca_discon_ind\(\)](#), [ng_l2cap_l2ca_discon_req\(\)](#), [ng_l2cap_l2ca_discon_rsp\(\)](#), [ng_](#)

l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lower_rcvmsg(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_cfm(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), ng_l2cap_process_discon_timeout(), ng_l2cap_send_hook_info(), ng_l2tp_rcvmsg(), ng_mppc_rcvmsg(), ng_nat_rcvmsg(), ng_netflow_flow_show(), ng_netflow_rcvmsg(), ng_one2many_rcvmsg(), ng_ppp_rcvmsg(), ng_pppoe_rcvmsg(), ng_pptpgre_rcvmsg(), ng_pred1_rcvmsg(), ng_rfc1490_rcvmsg(), ng_source_rcvmsg(), ng_source_set_autosrc(), ng_sppp_rcvmsg(), ng_sscfu_rcvmsg(), ng_sscop_rcvmsg(), ng_tag_rcvmsg(), ng_tcpmss_rcvmsg(), ng_ubt_rcvmsg(), ng_uni_rcvmsg(), ng_vjc_rcvmsg(), ng_vlan_rcvmsg(), ng_xxx_rcvmsg(), nga_rcvmsg(), ngh_rcvmsg(), nglmi_rcvmsg(), ngt_rcvmsg(), pppoe_send_event(), send_acname(), send_sessionid(), and sync_con_queue().

6.180.2.2 struct `ng_mesg::ng_msghdr` `ng_mesg::header`

Referenced by `cisco_rcvmsg()`, `flow_lower()`, `flow_upper()`, `ng_apply_item()`, `ng_atm_rcvmsg()`, `ng_atmllc_rcvmsg()`, `ng_atmpif_rcvmsg()`, `ng_bpf_rcvmsg()`, `ng_bridge_rcvmsg()`, `ng_bt3c_rcvmsg()`, `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_msg_input()`, `ng_btsocket_hci_raw_node_rcvmsg()`, `ng_btsocket_hci_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_l2ca_msg_input()`, `ng_btsocket_l2cap_node_rcvmsg()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_ccatm_rcvmsg()`, `ng_deflate_rcvmsg()`, `ng_device_rcvmsg()`, `ng_eiface_rcvmsg()`, `ng_etf_rcvmsg()`, `ng_ether_rcvmsg()`, `ng_fec_rcvmsg()`, `ng_generic_msg()`, `ng_gif_demux_rcvmsg()`, `ng_gif_rcvmsg()`, `ng_h4_rcvmsg()`, `ng_hci_default_rcvmsg()`, `ng_hci_upper_rcvmsg()`, `ng_iface_rcvmsg()`, `ng_ksocket_finish_accept()`, `ng_ksocket_rcvmsg()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, `ng_l2cap_lp_discon_ind()`, `ng_l2cap_lp_qos_cfm()`, `ng_l2cap_lp_qos_ind()`, `ng_l2cap_upper_rcvmsg()`, `ng_l2tp_rcvmsg()`, `ng_mppc_rcvmsg()`, `ng_nat_rcvmsg()`, `ng_netflow_rcvmsg()`, `ng_one2many_rcvmsg()`, `ng_package_msg()`, `ng_parse_ng_mesg_getLength()`, `ng_parse_sockoptval_getLength()`, `ng_ppp_rcvmsg()`, `ng_pppoe_rcvmsg()`, `ng_pptpgre_rcvmsg()`, `ng_pred1_rcvmsg()`, `ng_rfc1490_rcvmsg()`, `ng_source_rcvmsg()`, `ng_sppp_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_tag_rcvmsg()`, `ng_tcpmss_rcvmsg()`, `ng_ubt_rcvmsg()`, `ng_UI_rcvmsg()`, `ng_uni_rcvmsg()`, `ng_vjc_rcvmsg()`, `ng_vlan_rcvmsg()`, `ng_xxx_rcvmsg()`, `nga_rcvmsg()`, `nge_rcvmsg()`, `ng_h_rcvmsg()`, `nglmi_rcvmsg()`, `ngs_rcvmsg()`, and `ngt_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.181 ng_mesg::ng_msghdr Struct Reference

```
#include <ng_message.h>
```

Data Fields

- u_char [version](#)
- u_char [spare](#)
- u_int16_t [spare2](#)
- u_int32_t [arglen](#)
- u_int32_t [cmd](#)
- u_int32_t [flags](#)
- u_int32_t [token](#)
- u_int32_t [typecookie](#)
- u_char [cmdstr](#) [NG_CMDSTRSIZ]

6.181.1 Detailed Description

Definition at line 67 of file ng_message.h.

6.181.2 Field Documentation

6.181.2.1 u_int32_t ng_mesg::ng_msghdr::arglen

Definition at line 71 of file ng_message.h.

Referenced by [cisco_rcvmsg\(\)](#), [flow_lower\(\)](#), [flow_upper\(\)](#), [ng_atm_rcvmsg\(\)](#), [ng_atmpif_rcvmsg\(\)](#), [ng_bpf_rcvmsg\(\)](#), [ng_bridge_rcvmsg\(\)](#), [ng_bt3c_rcvmsg\(\)](#), [ng_btsocket_l2cap_default_msg_input\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_req_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_req_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_con_rsp_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_discon_ind\(\)](#), [ng_btsocket_l2cap_process_l2ca_discon_rsp\(\)](#), [ng_btsocket_l2cap_process_l2ca_write_rsp\(\)](#), [ng_btsocket_l2cap_raw_input\(\)](#), [ng_ccatm_rcvmsg\(\)](#), [ng_deflate_rcvmsg\(\)](#), [ng_eiface_rcvmsg\(\)](#), [ng_etc_rcvmsg\(\)](#), [ng_ether_rcvmsg\(\)](#), [ng_generic_msg\(\)](#), [ng_h4_rcvmsg\(\)](#), [ng_hci_default_rcvmsg\(\)](#), [ng_ksocket_rcvmsg\(\)](#), [ng_l2cap_default_rcvmsg\(\)](#), [ng_l2cap_l2ca_cfg_req\(\)](#), [ng_l2cap_l2ca_cfg_rsp_req\(\)](#), [ng_l2cap_l2ca_con_req\(\)](#), [ng_l2cap_l2ca_con_rsp_req\(\)](#), [ng_l2cap_l2ca_discon_req\(\)](#), [ng_l2cap_l2ca_enable_clt\(\)](#), [ng_l2cap_l2ca_get_info_req\(\)](#), [ng_l2cap_l2ca_ping_req\(\)](#), [ng_l2cap_lower_rcvmsg\(\)](#), [ng_l2cap_lp_con_cfm\(\)](#), [ng_l2cap_lp_con_ind\(\)](#), [ng_l2cap_lp_discon_ind\(\)](#), [ng_l2cap_lp_qos_cfm\(\)](#), [ng_l2cap_lp_qos_ind\(\)](#), [ng_l2tp_rcvmsg\(\)](#), [ng_mppc_rcvmsg\(\)](#), [ng_nat_rcvmsg\(\)](#), [ng_netflow_rcvmsg\(\)](#), [ng_one2many_rcvmsg\(\)](#), [ng_parse_ng_mesg_getLength\(\)](#), [ng_parse_sockoptval_getLength\(\)](#), [ng_ppp_rcvmsg\(\)](#), [ng_pppoe_rcvmsg\(\)](#), [ng_pptpgre_rcvmsg\(\)](#), [ng_pred1_rcvmsg\(\)](#), [ng_rfc1490_rcvmsg\(\)](#), [ng_source_rcvmsg\(\)](#), [ng_sscfu_rcvmsg\(\)](#), [ng_sscop_rcvmsg\(\)](#), [ng_tag_rcvmsg\(\)](#), [ng_tcpmss_rcvmsg\(\)](#), [ng_ubt_rcvmsg\(\)](#), [ng_uni_rcvmsg\(\)](#), [ng_vjc_rcvmsg\(\)](#), [ng_vlan_rcvmsg\(\)](#), [ng_XXX_rcvmsg\(\)](#), [nga_rcvmsg\(\)](#), [ngh_rcvmsg\(\)](#), [nglmi_rcvmsg\(\)](#), [ngs_rcvmsg\(\)](#), and [ngt_rcvmsg\(\)](#).

6.181.2.2 u_int32_t ng_mesg::ng_msghdr::cmd

Definition at line 72 of file ng_message.h.

Referenced by `cisco_rcvmsg()`, `flow_lower()`, `flow_upper()`, `ng_atm_rcvmsg()`, `ng_atmpif_rcvmsg()`, `ng_bpf_rcvmsg()`, `ng_bridge_rcvmsg()`, `ng_bt3c_rcvmsg()`, `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_l2ca_msg_input()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_ccatm_rcvmsg()`, `ng_deflate_rcvmsg()`, `ng_device_rcvmsg()`, `ng_eiface_rcvmsg()`, `ng_etf_rcvmsg()`, `ng_ether_rcvmsg()`, `ng_fec_rcvmsg()`, `ng_generic_msg()`, `ng_gif_demux_rcvmsg()`, `ng_gif_rcvmsg()`, `ng_h4_rcvmsg()`, `ng_hci_default_rcvmsg()`, `ng_hci_upper_rcvmsg()`, `ng_iface_rcvmsg()`, `ng_ksocket_rcvmsg()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_upper_rcvmsg()`, `ng_l2tp_rcvmsg()`, `ng_mppc_rcvmsg()`, `ng_nat_rcvmsg()`, `ng_netflow_rcvmsg()`, `ng_one2many_rcvmsg()`, `ng_package_msg()`, `ng_ppp_rcvmsg()`, `ng_pppoe_rcvmsg()`, `ng_pptpgre_rcvmsg()`, `ng_pred1_rcvmsg()`, `ng_rfc1490_rcvmsg()`, `ng_source_rcvmsg()`, `ng_sppp_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_tag_rcvmsg()`, `ng_tcpmss_rcvmsg()`, `ng_ubt_rcvmsg()`, `ng_uni_rcvmsg()`, `ng_vjc_rcvmsg()`, `ng_vlan_rcvmsg()`, `ng_xxx_rcvmsg()`, `nga_rcvmsg()`, `ngh_rcvmsg()`, `nglmi_rcvmsg()`, `ngs_rcvmsg()`, and `ngt_rcvmsg()`.

6.181.2.3 `u_char ng_mesg::ng_msghdr::cmdstr[NG_CMDSTRSIZ]`

Definition at line 76 of file `ng_message.h`.

Referenced by `ngs_rcvmsg()`.

6.181.2.4 `u_int32_t ng_mesg::ng_msghdr::flags`

Definition at line 73 of file `ng_message.h`.

Referenced by `cisco_rcvmsg()`, `ng_apply_item()`, `ng_atmllc_rcvmsg()`, `ng_btsocket_hci_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_ksocket_finish_accept()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_write_rsp()`, `ng_pppoe_rcvmsg()`, `ng_source_rcvmsg()`, `nge_rcvmsg()`, and `ngs_rcvmsg()`.

6.181.2.5 `u_char ng_mesg::ng_msghdr::spare`

Definition at line 69 of file `ng_message.h`.

6.181.2.6 `u_int16_t ng_mesg::ng_msghdr::spare2`

Definition at line 70 of file `ng_message.h`.

6.181.2.7 `u_int32_t ng_mesg::ng_msghdr::token`

Definition at line 74 of file `ng_message.h`.

Referenced by `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_msg_input()`, `ng_btsocket_hci_raw_node_rcvmsg()`, `ng_btsocket_hci_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_input()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, `ng_btsocket_l2cap_send_l2ca_discon_`

req(), ng_ksocket_finish_accept(), ng_ksocket_rcvmsg(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_cfg_rsp_rsp(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_con_rsp_rsp(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_write_rsp(), and ngs_rcvmsg().

6.181.2.8 u_int32_t ng_mesg::ng_msghdr::typecookie

Definition at line 75 of file ng_message.h.

Referenced by cisco_rcvmsg(), ng_apply_item(), ng_atm_rcvmsg(), ng_atmpif_rcvmsg(), ng_bpbf_rcvmsg(), ng_bridge_rcvmsg(), ng_bt3c_rcvmsg(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_ccatm_rcvmsg(), ng_deflate_rcvmsg(), ng_device_rcvmsg(), ng_eiface_rcvmsg(), ng_etf_rcvmsg(), ng_ether_rcvmsg(), ng_fec_rcvmsg(), ng_generic_msg(), ng_gif_demux_rcvmsg(), ng_gif_rcvmsg(), ng_h4_rcvmsg(), ng_hci_default_rcvmsg(), ng_hci_upper_rcvmsg(), ng_iface_rcvmsg(), ng_ksocket_rcvmsg(), ng_l2cap_default_rcvmsg(), ng_l2cap_lower_rcvmsg(), ng_l2cap_upper_rcvmsg(), ng_l2tp_rcvmsg(), ng_mppc_rcvmsg(), ng_nat_rcvmsg(), ng_netflow_rcvmsg(), ng_one2many_rcvmsg(), ng_ppp_rcvmsg(), ng_pppoe_rcvmsg(), ng_pptpgre_rcvmsg(), ng_pred1_rcvmsg(), ng_rfc1490_rcvmsg(), ng_source_rcvmsg(), ng_sppp_rcvmsg(), ng_sscfu_rcvmsg(), ng_sscop_rcvmsg(), ng_tag_rcvmsg(), ng_tcpmss_rcvmsg(), ng_ubt_rcvmsg(), ng_UI_rcvmsg(), ng_uni_rcvmsg(), ng_vjc_rcvmsg(), ng_vlan_rcvmsg(), ng_xxx_rcvmsg(), nga_rcvmsg(), ngh_rcvmsg(), nglmi_rcvmsg(), ngs_rcvmsg(), and ngt_rcvmsg().

6.181.2.9 u_char ng_mesg::ng_msghdr::version

Definition at line 68 of file ng_message.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.182 ng_mppc_config Struct Reference

```
#include <ng_mppc.h>
```

Data Fields

- u_char [enable](#)
- u_int32_t [bits](#)
- u_char [startkey](#) [MPPE_KEY_LEN]

6.182.1 Detailed Description

Definition at line 71 of file `ng_mppc.h`.

6.182.2 Field Documentation

6.182.2.1 u_int32_t [ng_mppc_config::bits](#)

Definition at line 73 of file `ng_mppc.h`.

Referenced by `ng_mppc_compress()`, `ng_mppc_decompress()`, and `ng_mppc_reset_req()`.

6.182.2.2 u_char [ng_mppc_config::enable](#)

Definition at line 72 of file `ng_mppc.h`.

6.182.2.3 u_char [ng_mppc_config::startkey](#)[MPPE_KEY_LEN]

Definition at line 74 of file `ng_mppc.h`.

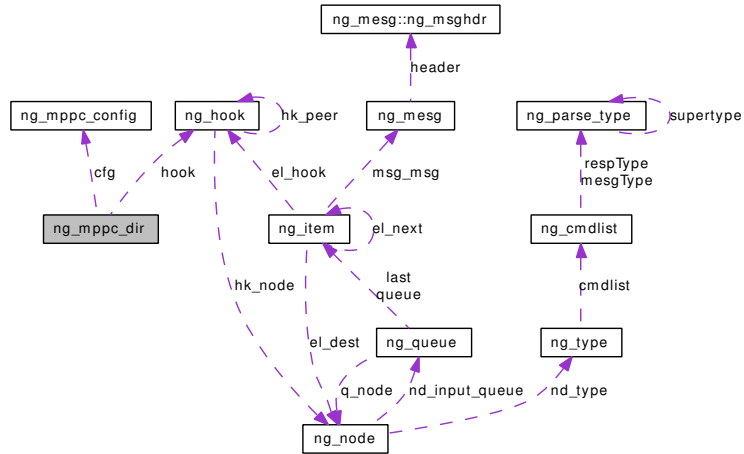
Referenced by `ng_mppc_compress()`, and `ng_mppc_decompress()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_mppc.h`

6.183 ng_mppc_dir Struct Reference

Collaboration diagram for ng_mppc_dir:



Data Fields

- [ng_mppc_config](#) `cfg`
- [hook_p](#) `hook`
- [u_int16_t](#) `cc:12`
- [u_char](#) `flushed`

6.183.1 Detailed Description

Definition at line 123 of file `ng_mppc.c`.

6.183.2 Field Documentation

6.183.2.1 `u_int16_t ng_mppc_dir::cc`

Definition at line 126 of file `ng_mppc.c`.

Referenced by `ng_mppc_compress()`, and `ng_mppc_decompress()`.

6.183.2.2 `struct ng_mppc_config ng_mppc_dir::cfg`

Definition at line 124 of file `ng_mppc.c`.

Referenced by `ng_mppc_compress()`, `ng_mppc_decompress()`, and `ng_mppc_reset_req()`.

6.183.2.3 `u_char ng_mppc_dir::flushed`

Definition at line 127 of file `ng_mppc.c`.

Referenced by `ng_mppc_compress()`, and `ng_mppc_reset_req()`.

6.183.2.4 [hook_p ng_mppc_dir::hook](#)

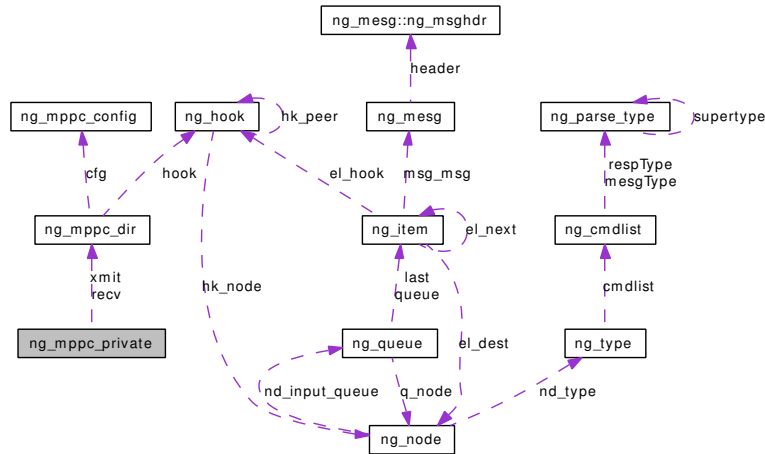
Definition at line 125 of file [ng_mppc.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_mppc.c](#)

6.184 ng_mppc_private Struct Reference

Collaboration diagram for ng_mppc_private:



Data Fields

- [ng_mppc_dir](#) xmit
- [ng_mppc_dir](#) recv
- [ng_ID_t](#) ctrlnode

6.184.1 Detailed Description

Definition at line 138 of file ng_mppc.c.

6.184.2 Field Documentation

6.184.2.1 [ng_ID_t](#) [ng_mppc_private::ctrlnode](#)

Definition at line 141 of file ng_mppc.c.

6.184.2.2 [struct](#) [ng_mppc_dir](#) [ng_mppc_private::recv](#)

Definition at line 140 of file ng_mppc.c.

6.184.2.3 [struct](#) [ng_mppc_dir](#) [ng_mppc_private::xmit](#)

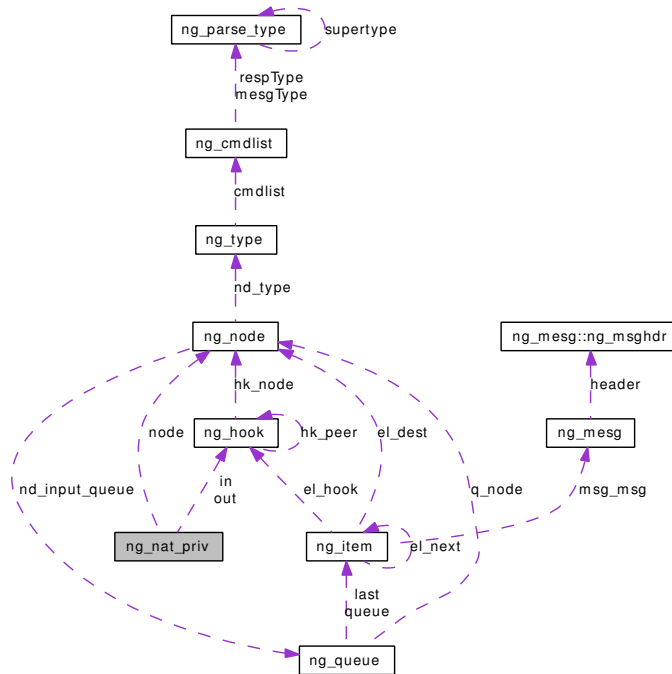
Definition at line 139 of file ng_mppc.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_mppc.c](#)

6.185 ng_nat_priv Struct Reference

Collaboration diagram for ng_nat_priv:



Data Fields

- `node_p` node
- `hook_p` in
- `hook_p` out
- libalias * lib
- `uint32_t` flags

6.185.1 Detailed Description

Definition at line 87 of file `ng_nat.c`.

6.185.2 Field Documentation

6.185.2.1 `uint32_t` `ng_nat_priv::flags`

Definition at line 92 of file `ng_nat.c`.

6.185.2.2 `hook_p` `ng_nat_priv::in`

Definition at line 89 of file `ng_nat.c`.

6.185.2.3 struct libalias* [ng_nat_priv::lib](#)

Definition at line 91 of file [ng_nat.c](#).

6.185.2.4 [node_p ng_nat_priv::node](#)

Definition at line 88 of file [ng_nat.c](#).

Referenced by [ng_nat_constructor\(\)](#), [ng_nat_newhook\(\)](#), and [ng_nat_rcvmsg\(\)](#).

6.185.2.5 [hook_p ng_nat_priv::out](#)

Definition at line 90 of file [ng_nat.c](#).

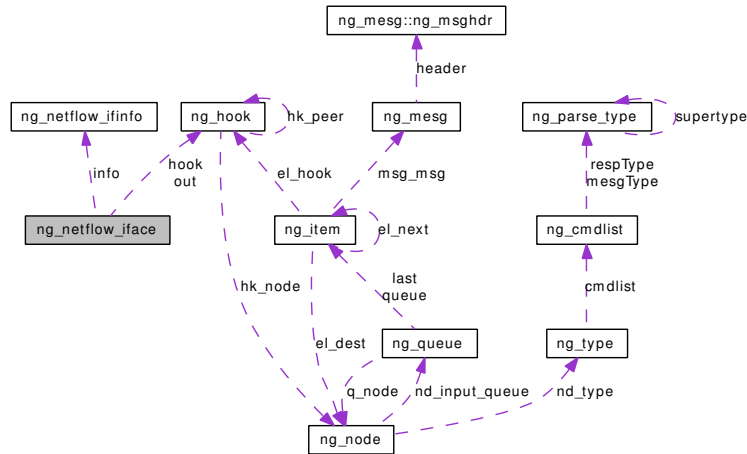
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_nat.c](#)

6.186 ng_netflow_iface Struct Reference

```
#include <ng_netflow.h>
```

Collaboration diagram for ng_netflow_iface:



Data Fields

- [hook_p](#) hook
- [hook_p](#) out
- [ng_netflow_ifinfo](#) info

6.186.1 Detailed Description

Definition at line 210 of file ng_netflow.h.

6.186.2 Field Documentation

6.186.2.1 [hook_p](#) ng_netflow_iface::hook

Definition at line 211 of file ng_netflow.h.

Referenced by [ng_netflow_disconnect\(\)](#), [ng_netflow_newhook\(\)](#), and [ng_netflow_rcvdata\(\)](#).

6.186.2.2 [struct ng_netflow_ifinfo](#) ng_netflow_iface::info

Definition at line 213 of file ng_netflow.h.

Referenced by [ng_netflow_flow_add\(\)](#), [ng_netflow_newhook\(\)](#), and [ng_netflow_rcvdata\(\)](#).

6.186.2.3 [hook_p](#) ng_netflow_iface::out

Definition at line 212 of file ng_netflow.h.

Referenced by [ng_netflow_disconnect\(\)](#), [ng_netflow_newhook\(\)](#), and [ng_netflow_rcvdata\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/ng_netflow.h](#)

6.187 ng_netflow_ifinfo Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- [uint32_t ifinfo_packets](#)
- [uint8_t ifinfo_dlt](#)
- [u_int16_t ifinfo_index](#)

6.187.1 Detailed Description

Definition at line 69 of file `ng_netflow.h`.

6.187.2 Field Documentation

6.187.2.1 [uint8_t ng_netflow_ifinfo::ifinfo_dlt](#)

Definition at line 71 of file `ng_netflow.h`.

Referenced by `ng_netflow_newhook()`, and `ng_netflow_rcvdata()`.

6.187.2.2 [u_int16_t ng_netflow_ifinfo::ifinfo_index](#)

Definition at line 73 of file `ng_netflow.h`.

Referenced by `ng_netflow_flow_add()`.

6.187.2.3 [uint32_t ng_netflow_ifinfo::ifinfo_packets](#)

Definition at line 70 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netflow/ng_netflow.h](#)

6.188 ng_netflow_info Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- [uint64_t ninfo_bytes](#)
- [uint32_t ninfo_packets](#)
- [uint32_t ninfo_used](#)
- [uint32_t ninfo_alloc_failed](#)
- [uint32_t ninfo_export_failed](#)
- [uint32_t ninfo_act_exp](#)
- [uint32_t ninfo_inact_exp](#)
- [uint32_t ninfo_inact_t](#)
- [uint32_t ninfo_act_t](#)

6.188.1 Detailed Description

Definition at line 56 of file ng_netflow.h.

6.188.2 Field Documentation

6.188.2.1 [uint32_t ng_netflow_info::ninfo_act_exp](#)

Definition at line 62 of file ng_netflow.h.

6.188.2.2 [uint32_t ng_netflow_info::ninfo_act_t](#)

Definition at line 65 of file ng_netflow.h.

6.188.2.3 [uint32_t ng_netflow_info::ninfo_alloc_failed](#)

Definition at line 60 of file ng_netflow.h.

6.188.2.4 [uint64_t ng_netflow_info::ninfo_bytes](#)

Definition at line 57 of file ng_netflow.h.

6.188.2.5 [uint32_t ng_netflow_info::ninfo_export_failed](#)

Definition at line 61 of file ng_netflow.h.

6.188.2.6 [uint32_t ng_netflow_info::ninfo_inact_exp](#)

Definition at line 63 of file ng_netflow.h.

6.188.2.7 `uint32_t ng_netflow_info::ninfo_inact_t`

Definition at line 64 of file `ng_netflow.h`.

6.188.2.8 `uint32_t ng_netflow_info::ninfo_packets`

Definition at line 58 of file `ng_netflow.h`.

6.188.2.9 `uint32_t ng_netflow_info::ninfo_used`

Definition at line 59 of file `ng_netflow.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.189 ng_netflow_setdlt Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- [uint16_t iface](#)
- [uint8_t dlt](#)

6.189.1 Detailed Description

Definition at line 78 of file `ng_netflow.h`.

6.189.2 Field Documentation

6.189.2.1 [uint8_t ng_netflow_setdlt::dlt](#)

Definition at line 80 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

6.189.2.2 [uint16_t ng_netflow_setdlt::iface](#)

Definition at line 79 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.190 ng_netflow_setifindex Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- [u_int16_t iface](#)
- [u_int16_t index](#)

6.190.1 Detailed Description

Definition at line 84 of file `ng_netflow.h`.

6.190.2 Field Documentation

6.190.2.1 [u_int16_t ng_netflow_setifindex::iface](#)

Definition at line 85 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

6.190.2.2 [u_int16_t ng_netflow_setifindex::index](#)

Definition at line 86 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.191 `ng_netflow_settimeouts` Struct Reference

```
#include <ng_netflow.h>
```

Data Fields

- `uint32_t` [inactive_timeout](#)
- `uint32_t` [active_timeout](#)

6.191.1 Detailed Description

Definition at line 90 of file `ng_netflow.h`.

6.191.2 Field Documentation

6.191.2.1 `uint32_t` [ng_netflow_settimeouts::active_timeout](#)

Definition at line 92 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

6.191.2.2 `uint32_t` [ng_netflow_settimeouts::inactive_timeout](#)

Definition at line 91 of file `ng_netflow.h`.

Referenced by `ng_netflow_rcvmsg()`.

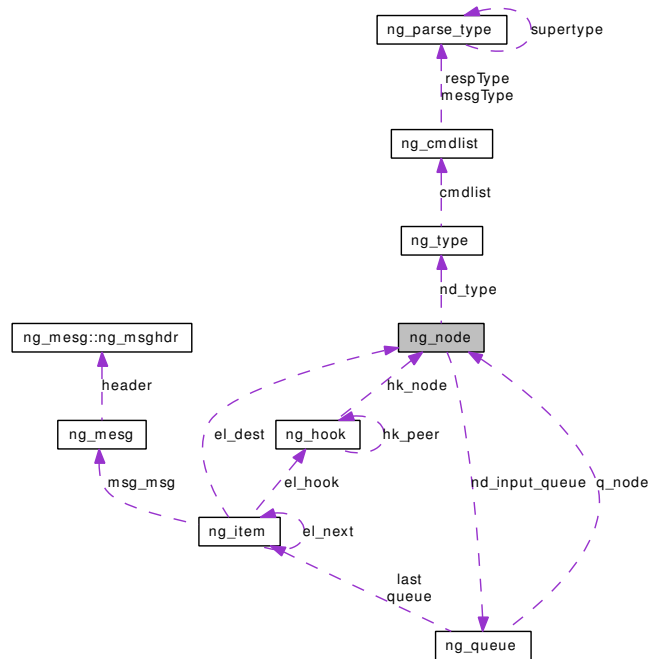
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.192 ng_node Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_node:



Public Member Functions

- [LIST_HEAD](#) (hooks, [ng_hook](#)) `nd_hooks`
- [LIST_ENTRY](#) ([ng_node](#)) `nd_nodes`
- [LIST_ENTRY](#) ([ng_node](#)) `nd_idnodes`
- [TAILQ_ENTRY](#) ([ng_node](#)) `nd_work`

Data Fields

- `char nd_name` [`NG_NODESIZ`]
- `ng_type * nd_type`
- `int nd_flags`
- `int nd_refs`
- `int nd_numhooks`
- `void * nd_private`
- `ng_ID_t nd_ID`
- `ng_queue nd_input_queue`

6.192.1 Detailed Description

Definition at line 330 of file `netgraph.h`.

6.192.2 Member Function Documentation

6.192.2.1 `ng_node::LIST_ENTRY` ([ng_node](#))

6.192.2.2 `ng_node::LIST_ENTRY` ([ng_node](#))

6.192.2.3 `ng_node::LIST_HEAD` ([hooks](#), [ng_hook](#))

6.192.2.4 `ng_node::TAILQ_ENTRY` ([ng_node](#))

6.192.3 Field Documentation

6.192.3.1 `int` [ng_node::nd_flags](#)

Definition at line 333 of file netgraph.h.

Referenced by `ng_atm_shutdown()`, `ng_ether_shutdown()`, `ng_gif_shutdown()`, `ng_rmnode()`, `ng_rmnode_self()`, `ng_setisr()`, `ng_snd_item()`, `ng_source_intr()`, `ng_source_rcvmsg()`, `ng_source_send()`, `ng_source_start()`, `ng_source_stop()`, `ng_worklist_remove()`, `ng_xxx_shutdown()`, and `ngintr()`.

6.192.3.2 `ng_ID_t` [ng_node::nd_ID](#)

Definition at line 337 of file netgraph.h.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, `ng_dequeue()`, `ng_make_node_common()`, `ng_pppoe_constructor()`, `ng_pppoe_disconnect()`, `ng_pppoe_newhook()`, `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, `ng_queue_rw()`, `ng_setisr()`, `ng_worklist_remove()`, `ngc_send()`, `ngintr()`, and `pppoe_ticker()`.

6.192.3.3 `struct` [ng_queue](#) [ng_node::nd_input_queue](#)

Definition at line 342 of file netgraph.h.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, `ng_apply_item()`, `ng_make_node_common()`, `ng_rmnode()`, `ng_setisr()`, `ng_snd_item()`, `ng_unref_node()`, `ng_worklist_remove()`, and `ngintr()`.

6.192.3.4 `char` [ng_node::nd_name](#)[NG_NODESIZ]

Definition at line 331 of file netgraph.h.

6.192.3.5 `int` [ng_node::nd_numhooks](#)

Definition at line 335 of file netgraph.h.

Referenced by `ng_add_hook()`, `ng_con_part2()`, `ng_destroy_hook()`, and `ng_generic_msg()`.

6.192.3.6 `void*` [ng_node::nd_private](#)

Definition at line 336 of file netgraph.h.

6.192.3.7 `int ng_node::nd_refs`

Definition at line 334 of file netgraph.h.

Referenced by `ng_unref_node()`.

6.192.3.8 `struct ng_type* ng_node::nd_type`

Definition at line 332 of file netgraph.h.

Referenced by `ng_add_hook()`, `ng_apply_item()`, `ng_con_part2()`, `ng_con_part3()`, `ng_connect_data()`, `ng_destroy_hook()`, `ng_findhook()`, `ng_generic_msg()`, `ng_h4_ioctl()`, `ng_make_node_common()`, `ng_mkpeer()`, `ng_rmnode()`, `ng_unref_node()`, `ngc_send()`, and `ngt_tioctl()`.

The documentation for this struct was generated from the following file:

- </usr/src/sys/netgraph/netgraph.h>

6.193 ng_one2many_config Struct Reference

```
#include <ng_one2many.h>
```

Data Fields

- `u_int32_t` [xmitAlg](#)
- `u_int32_t` [failAlg](#)
- `u_char` [enabledLinks](#) [NG_ONE2MANY_MAX_LINKS]

6.193.1 Detailed Description

Definition at line 70 of file `ng_one2many.h`.

6.193.2 Field Documentation

6.193.2.1 `u_char` [ng_one2many_config::enabledLinks](#)[NG_ONE2MANY_MAX_LINKS]

Definition at line 73 of file `ng_one2many.h`.

6.193.2.2 `u_int32_t` [ng_one2many_config::failAlg](#)

Definition at line 72 of file `ng_one2many.h`.

6.193.2.3 `u_int32_t` [ng_one2many_config::xmitAlg](#)

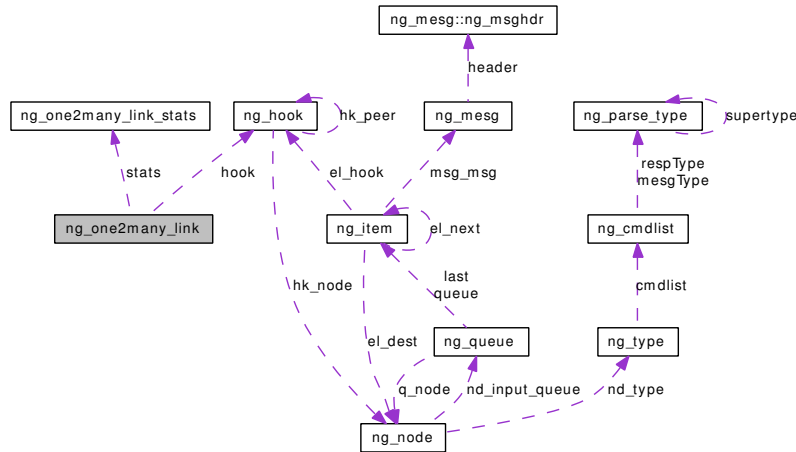
Definition at line 71 of file `ng_one2many.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_one2many.h`

6.194 ng_one2many_link Struct Reference

Collaboration diagram for ng_one2many_link:



Data Fields

- [hook_p hook](#)
- [ng_one2many_link_stats stats](#)

6.194.1 Detailed Description

Definition at line 65 of file `ng_one2many.c`.

6.194.2 Field Documentation

6.194.2.1 [hook_p ng_one2many_link::hook](#)

Definition at line 66 of file `ng_one2many.c`.

Referenced by `ng_one2many_disconnect()`, `ng_one2many_newhook()`, and `ng_one2many_rcvdata()`.

6.194.2.2 [struct ng_one2many_link_stats ng_one2many_link::stats](#)

Definition at line 67 of file `ng_one2many.c`.

Referenced by `ng_one2many_newhook()`, `ng_one2many_rcvdata()`, and `ng_one2many_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_one2many.c`

6.195 ng_one2many_link_stats Struct Reference

```
#include <ng_one2many.h>
```

Data Fields

- [u_int64_t recvOctets](#)
- [u_int64_t recvPackets](#)
- [u_int64_t xmitOctets](#)
- [u_int64_t xmitPackets](#)
- [u_int64_t memoryFailures](#)

6.195.1 Detailed Description

Definition at line 85 of file `ng_one2many.h`.

6.195.2 Field Documentation

6.195.2.1 [u_int64_t ng_one2many_link_stats::memoryFailures](#)

Definition at line 90 of file `ng_one2many.h`.

Referenced by `ng_one2many_rcvdata()`.

6.195.2.2 [u_int64_t ng_one2many_link_stats::recvOctets](#)

Definition at line 86 of file `ng_one2many.h`.

Referenced by `ng_one2many_rcvdata()`.

6.195.2.3 [u_int64_t ng_one2many_link_stats::recvPackets](#)

Definition at line 87 of file `ng_one2many.h`.

Referenced by `ng_one2many_rcvdata()`.

6.195.2.4 [u_int64_t ng_one2many_link_stats::xmitOctets](#)

Definition at line 88 of file `ng_one2many.h`.

Referenced by `ng_one2many_rcvdata()`.

6.195.2.5 [u_int64_t ng_one2many_link_stats::xmitPackets](#)

Definition at line 89 of file `ng_one2many.h`.

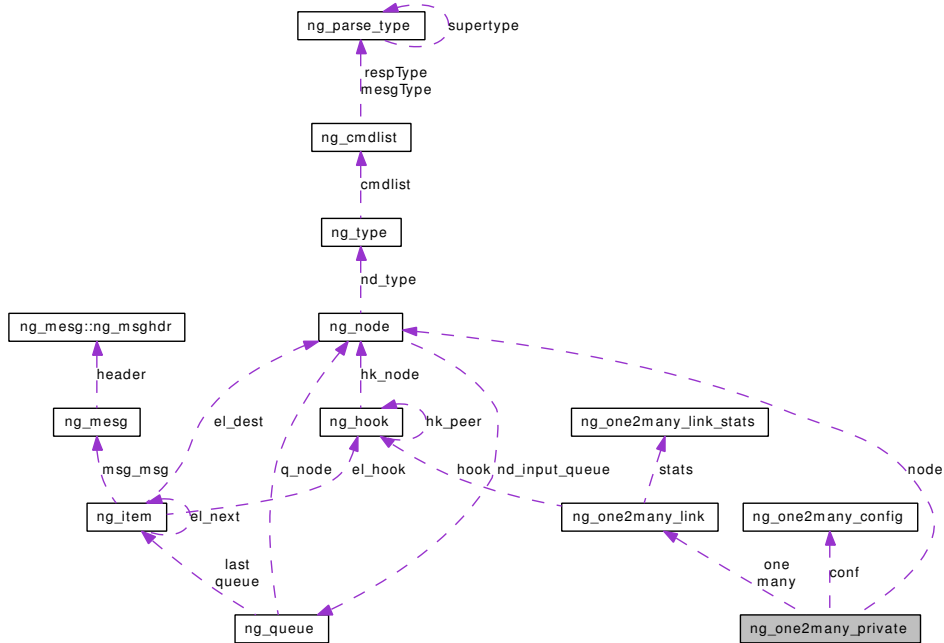
Referenced by `ng_one2many_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_one2many.h](#)

6.196 ng_one2many_private Struct Reference

Collaboration diagram for ng_one2many_private:



Data Fields

- `node_p` `node`
- `ng_one2many_config` `conf`
- `ng_one2many_link` `one`
- `ng_one2many_link` `many` [`NG_ONE2MANY_MAX_LINKS`]
- `u_int16_t` `nextMany`
- `u_int16_t` `numActiveMany`
- `u_int16_t` `activeMany` [`NG_ONE2MANY_MAX_LINKS`]

6.196.1 Detailed Description

Definition at line 71 of file `ng_one2many.c`.

6.196.2 Field Documentation

6.196.2.1 `u_int16_t` `ng_one2many_private::activeMany`[`NG_ONE2MANY_MAX_LINKS`]

Definition at line 78 of file `ng_one2many.c`.

6.196.2.2 `struct` `ng_one2many_config` `ng_one2many_private::conf`

Definition at line 73 of file `ng_one2many.c`.

6.196.2.3 struct [ng_one2many_link](#) [ng_one2many_private::many](#)[NG_ONE2MANY_MAX_LINKS]

Definition at line 75 of file [ng_one2many.c](#).

6.196.2.4 [u_int16_t](#) [ng_one2many_private::nextMany](#)

Definition at line 76 of file [ng_one2many.c](#).

6.196.2.5 [node_p](#) [ng_one2many_private::node](#)

Definition at line 72 of file [ng_one2many.c](#).

6.196.2.6 [u_int16_t](#) [ng_one2many_private::numActiveMany](#)

Definition at line 77 of file [ng_one2many.c](#).

6.196.2.7 struct [ng_one2many_link](#) [ng_one2many_private::one](#)

Definition at line 74 of file [ng_one2many.c](#).

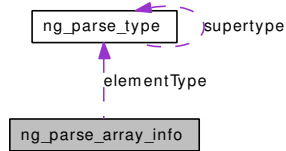
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_one2many.c](#)

6.197 ng_parse_array_info Struct Reference

```
#include <ng_parse.h>
```

Collaboration diagram for ng_parse_array_info:



Data Fields

- [ng_parse_type * elementType](#)
- [ng_parse_array_getLength_t * getLength](#)
- [ng_parse_array_getDefault_t * getDefault](#)

6.197.1 Detailed Description

Definition at line 363 of file ng_parse.h.

6.197.2 Field Documentation

6.197.2.1 struct [ng_parse_type*](#) [ng_parse_array_info::elementType](#)

Definition at line 364 of file ng_parse.h.

Referenced by [ng_array_getAlign\(\)](#), and [ng_get_composite_etype\(\)](#).

6.197.2.2 [ng_parse_array_getDefault_t*](#) [ng_parse_array_info::getDefault](#)

Definition at line 366 of file ng_parse.h.

Referenced by [ng_get_composite_elem_default\(\)](#).

6.197.2.3 [ng_parse_array_getLength_t*](#) [ng_parse_array_info::getLength](#)

Definition at line 365 of file ng_parse.h.

Referenced by [ng_get_composite_len\(\)](#).

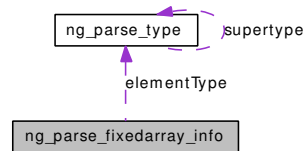
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.h](#)

6.198 ng_parse_fixedarray_info Struct Reference

```
#include <ng_parse.h>
```

Collaboration diagram for ng_parse_fixedarray_info:



Data Fields

- [ng_parse_type](#) * [elementType](#)
- [int](#) [length](#)
- [ng_parse_array_getDefault_t](#) * [getDefault](#)

6.198.1 Detailed Description

Definition at line 337 of file [ng_parse.h](#).

6.198.2 Field Documentation

6.198.2.1 struct [ng_parse_type](#)* [ng_parse_fixedarray_info::elementType](#)

Definition at line 338 of file [ng_parse.h](#).

Referenced by [ng_fixedarray_getAlign\(\)](#), and [ng_get_composite_etype\(\)](#).

6.198.2.2 [ng_parse_array_getDefault_t](#)* [ng_parse_fixedarray_info::getDefault](#)

Definition at line 340 of file [ng_parse.h](#).

Referenced by [ng_get_composite_elem_default\(\)](#).

6.198.2.3 int [ng_parse_fixedarray_info::length](#)

Definition at line 339 of file [ng_parse.h](#).

Referenced by [ng_get_composite_len\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.h](#)

6.199 ng_parse_fixedstring_info Struct Reference

```
#include <ng_parse.h>
```

Data Fields

- int [bufSize](#)

6.199.1 Detailed Description

Definition at line 390 of file `ng_parse.h`.

6.199.2 Field Documentation

6.199.2.1 int [ng_parse_fixedstring_info::bufSize](#)

Definition at line 391 of file `ng_parse.h`.

Referenced by `ng_fixedstring_getDefault()`, `ng_fixedstring_parse()`, and `ng_fixedstring_unparse()`.

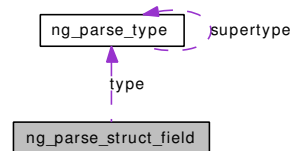
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.h](#)

6.200 ng_parse_struct_field Struct Reference

```
#include <ng_parse.h>
```

Collaboration diagram for ng_parse_struct_field:



Data Fields

- const char * [name](#)
- [ng_parse_type](#) * [type](#)
- int [alignment](#)

6.200.1 Detailed Description

Definition at line 310 of file [ng_parse.h](#).

6.200.2 Field Documentation

6.200.2.1 int [ng_parse_struct_field::alignment](#)

Definition at line 313 of file [ng_parse.h](#).

Referenced by [ng_parse_get_elem_pad\(\)](#).

6.200.2.2 const char* [ng_parse_struct_field::name](#)

Definition at line 311 of file [ng_parse.h](#).

Referenced by [ng_struct_getAlign\(\)](#).

6.200.2.3 struct [ng_parse_type](#)* [ng_parse_struct_field::type](#)

Definition at line 312 of file [ng_parse.h](#).

Referenced by [ng_get_composite_etype\(\)](#), [ng_get_composite_len\(\)](#), [ng_parse_composite\(\)](#), [ng_parse_get_elem_pad\(\)](#), [ng_struct_getAlign\(\)](#), and [ng_unparse_composite\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_parse.h](#)

6.201 ng_parse_type Struct Reference

```
#include <ng_parse.h>
```

Collaboration diagram for ng_parse_type:



Data Fields

- [ng_parse_type](#) * [supertype](#)
- const void * [info](#)
- void * [private](#)
- [ng_parse_t](#) * [parse](#)
- [ng_unparse_t](#) * [unparse](#)
- [ng_getDefault_t](#) * [getDefault](#)
- [ng_getAlign_t](#) * [getAlign](#)

6.201.1 Detailed Description

Definition at line 280 of file `ng_parse.h`.

6.201.2 Field Documentation

6.201.2.1 [ng_getAlign_t](#)* [ng_parse_type::getAlign](#)

Definition at line 287 of file `ng_parse.h`.

Referenced by `ng_get_getAlign_method()`.

6.201.2.2 [ng_getDefault_t](#)* [ng_parse_type::getDefault](#)

Definition at line 286 of file `ng_parse.h`.

Referenced by `ng_get_getDefault_method()`, and `ng_parse_getDefault()`.

6.201.2.3 `const void*` [ng_parse_type::info](#)

Definition at line 282 of file `ng_parse.h`.

Referenced by `ng_array_getAlign()`, `ng_bytearray_getDefault()`, `ng_bytearray_parse()`, `ng_bytearray_unparse()`, `ng_fixedarray_getAlign()`, `ng_fixedstring_getDefault()`, `ng_fixedstring_parse()`, `ng_fixedstring_unparse()`, `ng_get_composite_elem_default()`, `ng_get_composite_etype()`, `ng_get_composite_len()`, `ng_int16_unparse()`, `ng_int32_parse()`, `ng_int32_unparse()`, `ng_int64_unparse()`, `ng_int8_unparse()`, `ng_parse_composite()`, `ng_parse_get_elem_pad()`, `ng_struct_getAlign()`, and `ng_unparse_composite()`.

6.201.2.4 [ng_parse_t* ng_parse_type::parse](#)

Definition at line 284 of file `ng_parse.h`.

Referenced by `ng_get_parse_method()`, `ng_ksocket_sockaddr_parse()`, and `ng_parse()`.

6.201.2.5 [void* ng_parse_type::private](#)

Definition at line 283 of file `ng_parse.h`.

Referenced by `ng_bytearray_getDefault()`, `ng_bytearray_parse()`, `ng_bytearray_unparse()`, and `ng_parse_bytearray_subtype_getLength()`.

6.201.2.6 [struct ng_parse_type* ng_parse_type::supertype](#)

Definition at line 281 of file `ng_parse.h`.

Referenced by `ng_get_getAlign_method()`, `ng_get_getDefault_method()`, `ng_get_parse_method()`, `ng_get_unparse_method()`, `ng_ksocket_sockaddr_parse()`, and `ng_ksocket_sockaddr_unparse()`.

6.201.2.7 [ng_unparse_t* ng_parse_type::unparse](#)

Definition at line 285 of file `ng_parse.h`.

Referenced by `ng_get_unparse_method()`, `ng_ksocket_sockaddr_unparse()`, and `ng_unparse()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_parse.h`

6.202 ng_ppp_bund_conf Struct Reference

```
#include <ng_ppp.h>
```

Data Fields

- `u_int16_t` [mrru](#)
- `u_char` [enableMultilink](#)
- `u_char` [recvShortSeq](#)
- `u_char` [xmitShortSeq](#)
- `u_char` [enableRoundRobin](#)
- `u_char` [enableIP](#)
- `u_char` [enableIPv6](#)
- `u_char` [enableAtalk](#)
- `u_char` [enableIPX](#)
- `u_char` [enableCompression](#)
- `u_char` [enableDecompression](#)
- `u_char` [enableEncryption](#)
- `u_char` [enableDecryption](#)
- `u_char` [enableVJCompression](#)
- `u_char` [enableVJDecompression](#)

6.202.1 Detailed Description

Definition at line 139 of file `ng_ppp.h`.

6.202.2 Field Documentation

6.202.2.1 `u_char` [ng_ppp_bund_conf::enableAtalk](#)

Definition at line 147 of file `ng_ppp.h`.

6.202.2.2 `u_char` [ng_ppp_bund_conf::enableCompression](#)

Definition at line 149 of file `ng_ppp.h`.

6.202.2.3 `u_char` [ng_ppp_bund_conf::enableDecompression](#)

Definition at line 150 of file `ng_ppp.h`.

6.202.2.4 `u_char` [ng_ppp_bund_conf::enableDecryption](#)

Definition at line 152 of file `ng_ppp.h`.

6.202.2.5 `u_char` [ng_ppp_bund_conf::enableEncryption](#)

Definition at line 151 of file `ng_ppp.h`.

6.202.2.6 u_char ng_ppp_bund_conf::enableIP

Definition at line 145 of file ng_ppp.h.

6.202.2.7 u_char ng_ppp_bund_conf::enableIPv6

Definition at line 146 of file ng_ppp.h.

6.202.2.8 u_char ng_ppp_bund_conf::enableIPX

Definition at line 148 of file ng_ppp.h.

6.202.2.9 u_char ng_ppp_bund_conf::enableMultilink

Definition at line 141 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

6.202.2.10 u_char ng_ppp_bund_conf::enableRoundRobin

Definition at line 144 of file ng_ppp.h.

6.202.2.11 u_char ng_ppp_bund_conf::enableVJCompression

Definition at line 153 of file ng_ppp.h.

6.202.2.12 u_char ng_ppp_bund_conf::enableVJDecompression

Definition at line 154 of file ng_ppp.h.

6.202.2.13 u_int16_t ng_ppp_bund_conf::mrru

Definition at line 140 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

6.202.2.14 u_char ng_ppp_bund_conf::recvShortSeq

Definition at line 142 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

6.202.2.15 u_char ng_ppp_bund_conf::xmitShortSeq

Definition at line 143 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.h](#)

6.203 ng_ppp_frag Struct Reference

Data Fields

- int [seq](#)
- uint8_t [first](#)
- uint8_t [last](#)
- timeval [timestamp](#)
- mbuf * [data](#)

6.203.1 Detailed Description

Definition at line 184 of file ng_ppp.c.

6.203.2 Field Documentation

6.203.2.1 struct mbuf* [ng_ppp_frag::data](#)

Definition at line 189 of file ng_ppp.c.

Referenced by [ng_ppp_frag_checkstale\(\)](#), [ng_ppp_frag_process\(\)](#), [ng_ppp_frag_reset\(\)](#), [ng_ppp_frag_trim\(\)](#), [ng_ppp_get_packet\(\)](#), and [ng_ppp_mp_rcv\(\)](#).

6.203.2.2 uint8_t [ng_ppp_frag::first](#)

Definition at line 186 of file ng_ppp.c.

Referenced by [ng_ppp_check_packet\(\)](#), [ng_ppp_frag_checkstale\(\)](#), [ng_ppp_frag_trim\(\)](#), [ng_ppp_get_packet\(\)](#), and [ng_ppp_mp_rcv\(\)](#).

6.203.2.3 uint8_t [ng_ppp_frag::last](#)

Definition at line 187 of file ng_ppp.c.

Referenced by [ng_ppp_check_packet\(\)](#), [ng_ppp_frag_checkstale\(\)](#), [ng_ppp_frag_trim\(\)](#), [ng_ppp_get_packet\(\)](#), and [ng_ppp_mp_rcv\(\)](#).

6.203.2.4 int [ng_ppp_frag::seq](#)

Definition at line 185 of file ng_ppp.c.

Referenced by [ng_ppp_check_packet\(\)](#), [ng_ppp_frag_checkstale\(\)](#), [ng_ppp_frag_process\(\)](#), [ng_ppp_frag_trim\(\)](#), and [ng_ppp_mp_rcv\(\)](#).

6.203.2.5 struct timeval [ng_ppp_frag::timestamp](#)

Definition at line 188 of file ng_ppp.c.

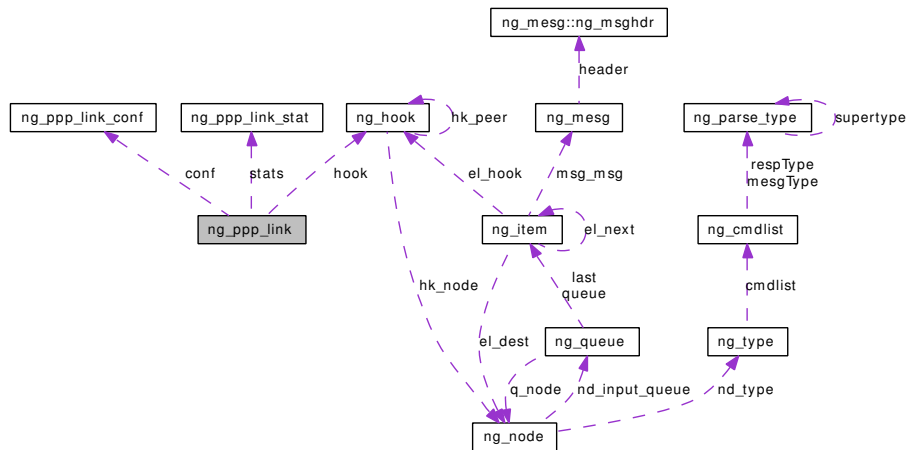
Referenced by [ng_ppp_frag_checkstale\(\)](#), and [ng_ppp_mp_rcv\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.c](#)

6.204 ng_ppp_link Struct Reference

Collaboration diagram for ng_ppp_link:



Data Fields

- [ng_ppp_link_conf](#) conf
- [ng_ppp_link_stat](#) stats
- [hook_p](#) hook
- [int32_t](#) seq
- [uint32_t](#) latency
- [timeval](#) lastWrite
- [int](#) bytesInQueue

6.204.1 Detailed Description

Definition at line 194 of file ng_ppp.c.

6.204.2 Field Documentation

6.204.2.1 `int ng_ppp_link::bytesInQueue`

Definition at line 201 of file ng_ppp.c.

Referenced by `ng_ppp_link_xmit()`.

6.204.2.2 `struct ng_ppp_link_conf ng_ppp_link::conf`

Definition at line 195 of file ng_ppp.c.

Referenced by `ng_ppp_link_xmit()`, `ng_ppp_mp_xmit()`, and `ng_ppp_rcvdata()`.

6.204.2.3 [hook_p ng_ppp_link::hook](#)

Definition at line 197 of file `ng_ppp.c`.

Referenced by `ng_ppp_link_xmit()`.

6.204.2.4 [struct timeval ng_ppp_link::lastWrite](#)

Definition at line 200 of file `ng_ppp.c`.

Referenced by `ng_ppp_link_xmit()`.

6.204.2.5 [uint32_t ng_ppp_link::latency](#)

Definition at line 199 of file `ng_ppp.c`.

6.204.2.6 [int32_t ng_ppp_link::seq](#)

Definition at line 198 of file `ng_ppp.c`.

Referenced by `ng_ppp_mp_recv()`.

6.204.2.7 [struct ng_ppp_link_stat ng_ppp_link::stats](#)

Definition at line 196 of file `ng_ppp.c`.

Referenced by `ng_ppp_link_xmit()`, `ng_ppp_mp_recv()`, and `ng_ppp_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.c](#)

6.205 ng_ppp_link_conf Struct Reference

```
#include <ng_ppp.h>
```

Data Fields

- u_char [enableLink](#)
- u_char [enableProtoComp](#)
- u_char [enableACFComp](#)
- u_int16_t [mru](#)
- u_int32_t [latency](#)
- u_int32_t [bandwidth](#)

6.205.1 Detailed Description

Definition at line 118 of file ng_ppp.h.

6.205.2 Field Documentation

6.205.2.1 u_int32_t [ng_ppp_link_conf::bandwidth](#)

Definition at line 124 of file ng_ppp.h.

Referenced by [ng_ppp_config_valid\(\)](#).

6.205.2.2 u_char [ng_ppp_link_conf::enableACFComp](#)

Definition at line 121 of file ng_ppp.h.

Referenced by [ng_ppp_link_xmit\(\)](#).

6.205.2.3 u_char [ng_ppp_link_conf::enableLink](#)

Definition at line 119 of file ng_ppp.h.

Referenced by [ng_ppp_config_valid\(\)](#), and [ng_ppp_rcvdata\(\)](#).

6.205.2.4 u_char [ng_ppp_link_conf::enableProtoComp](#)

Definition at line 120 of file ng_ppp.h.

Referenced by [ng_ppp_link_xmit\(\)](#).

6.205.2.5 u_int32_t [ng_ppp_link_conf::latency](#)

Definition at line 123 of file ng_ppp.h.

Referenced by [ng_ppp_config_valid\(\)](#).

6.205.2.6 `u_int16_t ng_ppp_link_conf::mru`

Definition at line 122 of file `ng_ppp.h`.

Referenced by `ng_ppp_config_valid()`, `ng_ppp_link_xmit()`, and `ng_ppp_mp_xmit()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.h](#)

6.206 ng_ppp_link_stat Struct Reference

```
#include <ng_ppp.h>
```

Data Fields

- [u_int32_t xmitFrames](#)
- [u_int32_t xmitOctets](#)
- [u_int32_t recvFrames](#)
- [u_int32_t recvOctets](#)
- [u_int32_t badProtos](#)
- [u_int32_t runs](#)
- [u_int32_t dupFragments](#)
- [u_int32_t dropFragments](#)

6.206.1 Detailed Description

Definition at line 191 of file ng_ppp.h.

6.206.2 Field Documentation

6.206.2.1 [u_int32_t ng_ppp_link_stat::badProtos](#)

Definition at line 196 of file ng_ppp.h.

Referenced by [ng_ppp_rcvdata\(\)](#).

6.206.2.2 [u_int32_t ng_ppp_link_stat::dropFragments](#)

Definition at line 199 of file ng_ppp.h.

Referenced by [ng_ppp_mp_recv\(\)](#).

6.206.2.3 [u_int32_t ng_ppp_link_stat::dupFragments](#)

Definition at line 198 of file ng_ppp.h.

Referenced by [ng_ppp_mp_recv\(\)](#).

6.206.2.4 [u_int32_t ng_ppp_link_stat::recvFrames](#)

Definition at line 194 of file ng_ppp.h.

Referenced by [ng_ppp_rcvdata\(\)](#).

6.206.2.5 [u_int32_t ng_ppp_link_stat::recvOctets](#)

Definition at line 195 of file ng_ppp.h.

Referenced by [ng_ppp_rcvdata\(\)](#).

6.206.2.6 `u_int32_t ng_ppp_link_stat::runs`

Definition at line 197 of file `ng_ppp.h`.

Referenced by `ng_ppp_mp_recv()`.

6.206.2.7 `u_int32_t ng_ppp_link_stat::xmitFrames`

Definition at line 192 of file `ng_ppp.h`.

Referenced by `ng_ppp_link_xmit()`.

6.206.2.8 `u_int32_t ng_ppp_link_stat::xmitOctets`

Definition at line 193 of file `ng_ppp.h`.

Referenced by `ng_ppp_link_xmit()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_ppp.h`

6.207 ng_ppp_mp_state Struct Reference

```
#include <ng_ppp.h>
```

Data Fields

- `int32_t rseq` [NG_PPP_MAX_LINKS]
- `int32_t mseq`
- `int32_t xseq`

6.207.1 Detailed Description

Definition at line 103 of file `ng_ppp.h`.

6.207.2 Field Documentation

6.207.2.1 `int32_t ng_ppp_mp_state::mseq`

Definition at line 105 of file `ng_ppp.h`.

6.207.2.2 `int32_t ng_ppp_mp_state::rseq`[NG_PPP_MAX_LINKS]

Definition at line 104 of file `ng_ppp.h`.

6.207.2.3 `int32_t ng_ppp_mp_state::xseq`

Definition at line 106 of file `ng_ppp.h`.

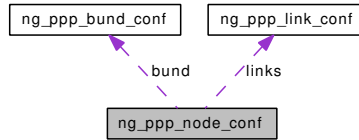
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_ppp.h`

6.208 ng_ppp_node_conf Struct Reference

```
#include <ng_ppp.h>
```

Collaboration diagram for ng_ppp_node_conf:



Data Fields

- [ng_ppp_bund_conf](#) bund
- [ng_ppp_link_conf](#) links [NG_PPP_MAX_LINKS]

6.208.1 Detailed Description

Definition at line 178 of file ng_ppp.h.

6.208.2 Field Documentation

6.208.2.1 struct [ng_ppp_bund_conf](#) [ng_ppp_node_conf::bund](#)

Definition at line 179 of file ng_ppp.h.

Referenced by [ng_ppp_config_valid\(\)](#).

6.208.2.2 struct [ng_ppp_link_conf](#) [ng_ppp_node_conf::links](#)[NG_PPP_MAX_LINKS]

Definition at line 180 of file ng_ppp.h.

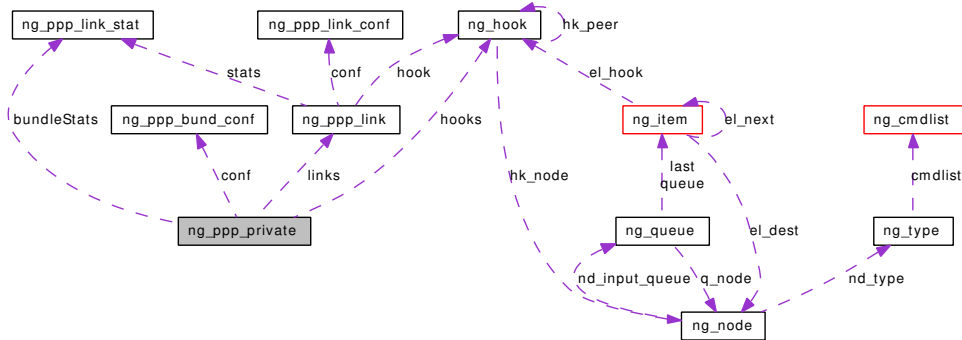
Referenced by [ng_ppp_config_valid\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.h](#)

6.209 ng_ppp_private Struct Reference

Collaboration diagram for ng_ppp_private:



Data Fields

- [ng_ppp_bund_conf](#) `conf`
- [ng_ppp_link_stat](#) `bundleStats`
- [ng_ppp_link](#) `links` [NG_PPP_MAX_LINKS]
- `int32_t` `xseq`
- `int32_t` `mseq`
- `uint16_t` `activeLinks` [NG_PPP_MAX_LINKS]
- `uint16_t` `numActiveLinks`
- `uint16_t` `lastLink`
- `uint8_t` `vjCompHooked`
- `uint8_t` `allLinksEqual`
- `hook_p` `hooks` [HOOK_INDEX_MAX]

6.209.1 Detailed Description

Definition at line 205 of file `ng_ppp.c`.

6.209.2 Field Documentation

6.209.2.1 `uint16_t` [ng_ppp_private::activeLinks](#)[NG_PPP_MAX_LINKS]

Definition at line 211 of file `ng_ppp.c`.

6.209.2.2 `uint8_t` [ng_ppp_private::allLinksEqual](#)

Definition at line 215 of file `ng_ppp.c`.

6.209.2.3 `struct` [ng_ppp_link_stat](#) [ng_ppp_private::bundleStats](#)

Definition at line 207 of file `ng_ppp.c`.

6.209.2.4 struct [ng_ppp_bund_conf](#) [ng_ppp_private::conf](#)

Definition at line 206 of file [ng_ppp.c](#).

6.209.2.5 hook_p [ng_ppp_private::hooks](#)[HOOK_INDEX_MAX]

Definition at line 216 of file [ng_ppp.c](#).

6.209.2.6 uint16_t [ng_ppp_private::lastLink](#)

Definition at line 213 of file [ng_ppp.c](#).

6.209.2.7 struct [ng_ppp_link](#) [ng_ppp_private::links](#)[NG_PPP_MAX_LINKS]

Definition at line 208 of file [ng_ppp.c](#).

6.209.2.8 int32_t [ng_ppp_private::mseq](#)

Definition at line 210 of file [ng_ppp.c](#).

6.209.2.9 uint16_t [ng_ppp_private::numActiveLinks](#)

Definition at line 212 of file [ng_ppp.c](#).

6.209.2.10 uint8_t [ng_ppp_private::vjCompHooked](#)

Definition at line 214 of file [ng_ppp.c](#).

6.209.2.11 int32_t [ng_ppp_private::xseq](#)

Definition at line 209 of file [ng_ppp.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ppp.c](#)

6.210 ng_pptpgre_ackp Struct Reference

Data Fields

- `int32_t` [ato](#)
- `int32_t` [rtt](#)
- `int32_t` [dev](#)
- `u_int16_t` [xmitWin](#)
- `callout` [sackTimer](#)
- `callout` [rackTimer](#)
- `u_int32_t` [winAck](#)
- `pptptime_t` [timeSent](#) [PPTP_XMIT_WIN]

6.210.1 Detailed Description

Definition at line 143 of file `ng_pptpgre.c`.

6.210.2 Field Documentation

6.210.2.1 `int32_t` [ng_pptpgre_ackp::ato](#)

Definition at line 144 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_rcv()`, `ng_pptpgre_rcv_ack_timeout()`, `ng_pptpgre_reset()`, and `ng_pptpgre_start_rcv_ack_timer()`.

6.210.2.2 `int32_t` [ng_pptpgre_ackp::dev](#)

Definition at line 146 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_rcv()`, `ng_pptpgre_rcv_ack_timeout()`, and `ng_pptpgre_reset()`.

6.210.2.3 `struct callout` [ng_pptpgre_ackp::rackTimer](#)

Definition at line 149 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_start_rcv_ack_timer()`, and `ng_pptpgre_stop_rcv_ack_timer()`.

6.210.2.4 `int32_t` [ng_pptpgre_ackp::rtt](#)

Definition at line 145 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_rcv()`, `ng_pptpgre_rcv_ack_timeout()`, and `ng_pptpgre_reset()`.

6.210.2.5 `struct callout` [ng_pptpgre_ackp::sackTimer](#)

Definition at line 148 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_rcv()`, `ng_pptpgre_start_send_ack_timer()`, and `ng_pptpgre_stop_send_ack_timer()`.

6.210.2.6 `pptptime_t ng_pptpgre_ackp::timeSent[PPTP_XMIT_WIN]`

Definition at line 151 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_recv()`, `ng_pptpgre_recv_ack_timeout()`, `ng_pptpgre_start_recv_ack_timer()`, and `ng_pptpgre_xmit()`.

6.210.2.7 `u_int32_t ng_pptpgre_ackp::winAck`

Definition at line 150 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_recv()`, `ng_pptpgre_recv_ack_timeout()`, and `ng_pptpgre_reset()`.

6.210.2.8 `u_int16_t ng_pptpgre_ackp::xmitWin`

Definition at line 147 of file `ng_pptpgre.c`.

Referenced by `ng_pptpgre_recv()`, `ng_pptpgre_recv_ack_timeout()`, `ng_pptpgre_reset()`, and `ng_pptpgre_xmit()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_pptpgre.c`

6.211 ng_pptpgre_conf Struct Reference

```
#include <ng_pptpgre.h>
```

Data Fields

- u_char [enabled](#)
- u_char [enableDelayedAck](#)
- u_char [enableAlwaysAck](#)
- u_char [enableWindowing](#)
- u_int16_t [cid](#)
- u_int16_t [peerCid](#)
- u_int16_t [recvWin](#)
- u_int16_t [peerPpd](#)

6.211.1 Detailed Description

Definition at line 56 of file ng_pptpgre.h.

6.211.2 Field Documentation

6.211.2.1 u_int16_t ng_pptpgre_conf::cid

Definition at line 61 of file ng_pptpgre.h.

6.211.2.2 u_char ng_pptpgre_conf::enableAlwaysAck

Definition at line 59 of file ng_pptpgre.h.

6.211.2.3 u_char ng_pptpgre_conf::enabled

Definition at line 57 of file ng_pptpgre.h.

6.211.2.4 u_char ng_pptpgre_conf::enableDelayedAck

Definition at line 58 of file ng_pptpgre.h.

6.211.2.5 u_char ng_pptpgre_conf::enableWindowing

Definition at line 60 of file ng_pptpgre.h.

6.211.2.6 u_int16_t ng_pptpgre_conf::peerCid

Definition at line 62 of file ng_pptpgre.h.

6.211.2.7 `u_int16_t ng_pptpgre_conf::peerPpd`

Definition at line 64 of file `ng_pptpgre.h`.

6.211.2.8 `u_int16_t ng_pptpgre_conf::recvWin`

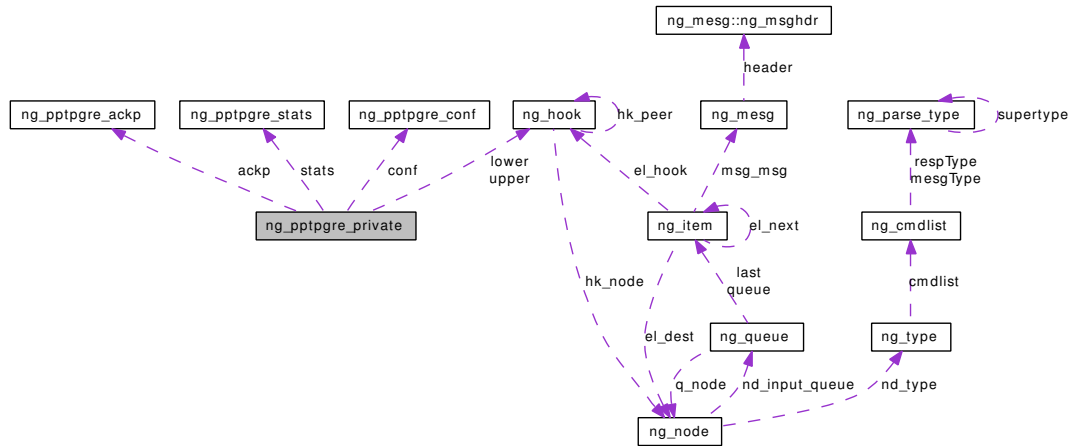
Definition at line 63 of file `ng_pptpgre.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pptpgre.h](#)

6.212 ng_pptpgre_private Struct Reference

Collaboration diagram for ng_pptpgre_private:



Data Fields

- [hook_p_upper](#)
- [hook_p_lower](#)
- [ng_pptpgre_conf conf](#)
- [ng_pptpgre_ackp ackp](#)
- [u_int32_t recvSeq](#)
- [u_int32_t xmitSeq](#)
- [u_int32_t recvAck](#)
- [u_int32_t xmitAck](#)
- [timeval startTime](#)
- [ng_pptpgre_stats stats](#)
- [mtx mtx](#)

6.212.1 Detailed Description

Definition at line 159 of file ng_pptpgre.c.

6.212.2 Field Documentation

6.212.2.1 struct [ng_pptpgre_ackp](#) [ng_pptpgre_private::ackp](#)

Definition at line 163 of file ng_pptpgre.c.

6.212.2.2 struct [ng_pptpgre_conf](#) [ng_pptpgre_private::conf](#)

Definition at line 162 of file ng_pptpgre.c.

6.212.2.3 [hook_p ng_pptpgre_private::lower](#)

Definition at line 161 of file [ng_pptpgre.c](#).

6.212.2.4 [struct mtx ng_pptpgre_private::mtx](#)

Definition at line 170 of file [ng_pptpgre.c](#).

6.212.2.5 [u_int32_t ng_pptpgre_private::recvAck](#)

Definition at line 166 of file [ng_pptpgre.c](#).

6.212.2.6 [u_int32_t ng_pptpgre_private::recvSeq](#)

Definition at line 164 of file [ng_pptpgre.c](#).

6.212.2.7 [struct timeval ng_pptpgre_private::startTime](#)

Definition at line 168 of file [ng_pptpgre.c](#).

6.212.2.8 [struct ng_pptpgre_stats ng_pptpgre_private::stats](#)

Definition at line 169 of file [ng_pptpgre.c](#).

6.212.2.9 [hook_p ng_pptpgre_private::upper](#)

Definition at line 160 of file [ng_pptpgre.c](#).

6.212.2.10 [u_int32_t ng_pptpgre_private::xmitAck](#)

Definition at line 167 of file [ng_pptpgre.c](#).

6.212.2.11 [u_int32_t ng_pptpgre_private::xmitSeq](#)

Definition at line 165 of file [ng_pptpgre.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pptpgre.c](#)

6.213 ng_pptpgre_stats Struct Reference

```
#include <ng_pptpgre.h>
```

Data Fields

- [u_int32_t xmitPackets](#)
- [u_int32_t xmitOctets](#)
- [u_int32_t xmitLoneAcks](#)
- [u_int32_t xmitDrops](#)
- [u_int32_t xmitTooBig](#)
- [u_int32_t recvPackets](#)
- [u_int32_t recvOctets](#)
- [u_int32_t recvRunts](#)
- [u_int32_t recvBadGRE](#)
- [u_int32_t recvBadAcks](#)
- [u_int32_t recvBadCID](#)
- [u_int32_t recvOutOfOrder](#)
- [u_int32_t recvDuplicates](#)
- [u_int32_t recvLoneAcks](#)
- [u_int32_t recvAckTimeouts](#)
- [u_int32_t memoryFailures](#)

6.213.1 Detailed Description

Definition at line 82 of file `ng_pptpgre.h`.

6.213.2 Field Documentation

6.213.2.1 [u_int32_t ng_pptpgre_stats::memoryFailures](#)

Definition at line 98 of file `ng_pptpgre.h`.

6.213.2.2 [u_int32_t ng_pptpgre_stats::recvAckTimeouts](#)

Definition at line 97 of file `ng_pptpgre.h`.

6.213.2.3 [u_int32_t ng_pptpgre_stats::recvBadAcks](#)

Definition at line 92 of file `ng_pptpgre.h`.

6.213.2.4 [u_int32_t ng_pptpgre_stats::recvBadCID](#)

Definition at line 93 of file `ng_pptpgre.h`.

6.213.2.5 [u_int32_t ng_pptpgre_stats::recvBadGRE](#)

Definition at line 91 of file `ng_pptpgre.h`.

6.213.2.6 [u_int32_t ng_pptpgre_stats::recvDuplicates](#)

Definition at line 95 of file [ng_pptpgre.h](#).

6.213.2.7 [u_int32_t ng_pptpgre_stats::recvLoneAcks](#)

Definition at line 96 of file [ng_pptpgre.h](#).

6.213.2.8 [u_int32_t ng_pptpgre_stats::recvOctets](#)

Definition at line 89 of file [ng_pptpgre.h](#).

6.213.2.9 [u_int32_t ng_pptpgre_stats::recvOutOfOrder](#)

Definition at line 94 of file [ng_pptpgre.h](#).

6.213.2.10 [u_int32_t ng_pptpgre_stats::recvPackets](#)

Definition at line 88 of file [ng_pptpgre.h](#).

6.213.2.11 [u_int32_t ng_pptpgre_stats::recvRunts](#)

Definition at line 90 of file [ng_pptpgre.h](#).

6.213.2.12 [u_int32_t ng_pptpgre_stats::xmitDrops](#)

Definition at line 86 of file [ng_pptpgre.h](#).

6.213.2.13 [u_int32_t ng_pptpgre_stats::xmitLoneAcks](#)

Definition at line 85 of file [ng_pptpgre.h](#).

6.213.2.14 [u_int32_t ng_pptpgre_stats::xmitOctets](#)

Definition at line 84 of file [ng_pptpgre.h](#).

6.213.2.15 [u_int32_t ng_pptpgre_stats::xmitPackets](#)

Definition at line 83 of file [ng_pptpgre.h](#).

6.213.2.16 [u_int32_t ng_pptpgre_stats::xmitTooBig](#)

Definition at line 87 of file [ng_pptpgre.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pptpgre.h](#)

6.214 ng_pred1_config Struct Reference

```
#include <ng_pred1.h>
```

Data Fields

- u_char [enable](#)

6.214.1 Detailed Description

Definition at line 42 of file ng_pred1.h.

6.214.2 Field Documentation

6.214.2.1 u_char [ng_pred1_config::enable](#)

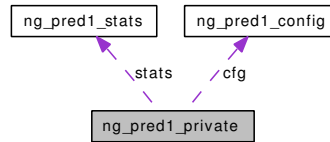
Definition at line 43 of file ng_pred1.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pred1.h](#)

6.215 ng_pred1_private Struct Reference

Collaboration diagram for ng_pred1_private:



Data Fields

- [ng_pred1_config](#) `cfg`
- `u_char` [GuessTable](#) [PRED1_TABLE_SIZE]
- `u_char` [inbuf](#) [PRED1_BUF_SIZE]
- `u_char` [outbuf](#) [PRED1_BUF_SIZE]
- [ng_pred1_stats](#) `stats`
- `uint16_t` [Hash](#)
- `ng_ID_t` [ctrlnode](#)
- `uint16_t` [seqnum](#)
- `u_char` [compress](#)

6.215.1 Detailed Description

Definition at line 70 of file `ng_pred1.c`.

6.215.2 Field Documentation

6.215.2.1 struct [ng_pred1_config](#) `ng_pred1_private::cfg`

Definition at line 71 of file `ng_pred1.c`.

6.215.2.2 `u_char` [ng_pred1_private::compress](#)

Definition at line 79 of file `ng_pred1.c`.

6.215.2.3 `ng_ID_t` [ng_pred1_private::ctrlnode](#)

Definition at line 77 of file `ng_pred1.c`.

6.215.2.4 `u_char` [ng_pred1_private::GuessTable](#)[PRED1_TABLE_SIZE]

Definition at line 72 of file `ng_pred1.c`.

6.215.2.5 `uint16_t` [ng_pred1_private::Hash](#)

Definition at line 76 of file `ng_pred1.c`.

6.215.2.6 `u_char ng_pred1_private::inbuf[PRED1_BUF_SIZE]`

Definition at line 73 of file `ng_pred1.c`.

6.215.2.7 `u_char ng_pred1_private::outbuf[PRED1_BUF_SIZE]`

Definition at line 74 of file `ng_pred1.c`.

6.215.2.8 `uint16_t ng_pred1_private::seqnum`

Definition at line 78 of file `ng_pred1.c`.

6.215.2.9 `struct ng_pred1_stats ng_pred1_private::stats`

Definition at line 75 of file `ng_pred1.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_pred1.c`

6.216 ng_pred1_stats Struct Reference

```
#include <ng_pred1.h>
```

Data Fields

- [uint64_t FramesPlain](#)
- [uint64_t FramesComp](#)
- [uint64_t FramesUncomp](#)
- [uint64_t InOctets](#)
- [uint64_t OutOctets](#)
- [uint64_t Errors](#)

6.216.1 Detailed Description

Definition at line 53 of file `ng_pred1.h`.

6.216.2 Field Documentation

6.216.2.1 [uint64_t ng_pred1_stats::Errors](#)

Definition at line 59 of file `ng_pred1.h`.

6.216.2.2 [uint64_t ng_pred1_stats::FramesComp](#)

Definition at line 55 of file `ng_pred1.h`.

6.216.2.3 [uint64_t ng_pred1_stats::FramesPlain](#)

Definition at line 54 of file `ng_pred1.h`.

6.216.2.4 [uint64_t ng_pred1_stats::FramesUncomp](#)

Definition at line 56 of file `ng_pred1.h`.

6.216.2.5 [uint64_t ng_pred1_stats::InOctets](#)

Definition at line 57 of file `ng_pred1.h`.

6.216.2.6 [uint64_t ng_pred1_stats::OutOctets](#)

Definition at line 58 of file `ng_pred1.h`.

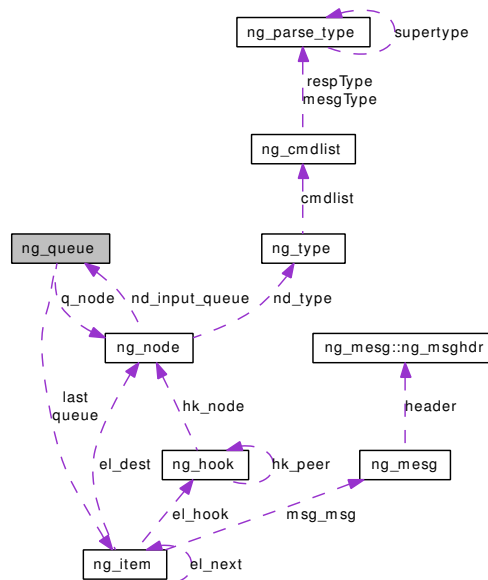
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pred1.h](#)

6.217 ng_queue Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_queue:



Data Fields

- u_long [q_flags](#)
- mtx [q_mtx](#)
- [item_p](#) queue
- [item_p](#) * [last](#)
- [ng_node](#) * [q_node](#)

6.217.1 Detailed Description

Definition at line 322 of file netgraph.h.

6.217.2 Field Documentation

6.217.2.1 [item_p](#)* [ng_queue::last](#)

Definition at line 326 of file netgraph.h.

Referenced by [ng_dequeue\(\)](#), [ng_flush_input_queue\(\)](#), [ng_make_node_common\(\)](#), and [ng_queue_rw\(\)](#).

6.217.2.2 u_long [ng_queue::q_flags](#)

Definition at line 323 of file netgraph.h.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, `ng_dequeue()`, `ng_flush_input_queue()`, `ng_leave_read()`, `ng_leave_write()`, `ng_make_node_common()`, and `ng_queue_rw()`.

6.217.2.3 `struct mtx ng_queue::q_mtx`

Definition at line 324 of file `netgraph.h`.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, `ng_dequeue()`, `ng_flush_input_queue()`, `ng_make_node_common()`, `ng_queue_rw()`, `ng_setisr()`, `ng_snd_item()`, `ng_unref_node()`, `ng_worklist_remove()`, and `ngintr()`.

6.217.2.4 `struct ng_node* ng_queue::q_node`

Definition at line 327 of file `netgraph.h`.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, `ng_dequeue()`, `ng_flush_input_queue()`, `ng_make_node_common()`, `ng_queue_rw()`, and `ng_snd_item()`.

6.217.2.5 `item_p ng_queue::queue`

Definition at line 325 of file `netgraph.h`.

Referenced by `ng_dequeue()`, `ng_flush_input_queue()`, `ng_make_node_common()`, and `ng_queue_rw()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netgraph.h`

6.218 ng_rfc1490_encap_t Struct Reference

Data Fields

- [u_int8_t method](#)
- `const char * name`

6.218.1 Detailed Description

Definition at line 92 of file `ng_rfc1490.c`.

6.218.2 Field Documentation

6.218.2.1 [u_int8_t ng_rfc1490_encap_t::method](#)

Definition at line 93 of file `ng_rfc1490.c`.

Referenced by `ng_rfc1490_rcvmsg()`.

6.218.2.2 `const char*` [ng_rfc1490_encap_t::name](#)

Definition at line 94 of file `ng_rfc1490.c`.

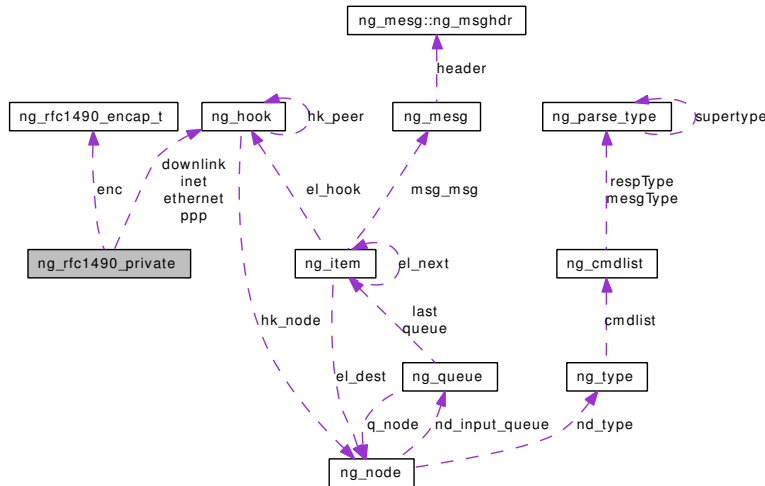
Referenced by `ng_rfc1490_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_rfc1490.c`

6.219 ng_rfc1490_private Struct Reference

Collaboration diagram for ng_rfc1490_private:



Data Fields

- `hook_p` `downlink`
- `hook_p` `ppp`
- `hook_p` `inet`
- `hook_p` `ethernet`
- `ng_rfc1490_encap_t` * `enc`

6.219.1 Detailed Description

Definition at line 105 of file `ng_rfc1490.c`.

6.219.2 Field Documentation

6.219.2.1 `hook_p` `ng_rfc1490_private::downlink`

Definition at line 106 of file `ng_rfc1490.c`.

6.219.2.2 `struct ng_rfc1490_encap_t*` `ng_rfc1490_private::enc`

Definition at line 110 of file `ng_rfc1490.c`.

6.219.2.3 `hook_p` `ng_rfc1490_private::ethernet`

Definition at line 109 of file `ng_rfc1490.c`.

6.219.2.4 `hook_p ng_rfc1490_private::inet`

Definition at line 108 of file `ng_rfc1490.c`.

6.219.2.5 `hook_p ng_rfc1490_private::ppp`

Definition at line 107 of file `ng_rfc1490.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_rfc1490.c](#)

6.220 ng_source_stats Struct Reference

```
#include <ng_source.h>
```

Data Fields

- uint64_t [outOctets](#)
- uint64_t [outFrames](#)
- uint32_t [queueOctets](#)
- uint32_t [queueFrames](#)
- uint32_t [maxPps](#)
- timeval [startTime](#)
- timeval [endTime](#)
- timeval [elapsedTime](#)
- timeval [lastTime](#)

6.220.1 Detailed Description

Definition at line 54 of file `ng_source.h`.

6.220.2 Field Documentation

6.220.2.1 struct timeval [ng_source_stats::elapsedTime](#)

Definition at line 62 of file `ng_source.h`.

6.220.2.2 struct timeval [ng_source_stats::endTime](#)

Definition at line 61 of file `ng_source.h`.

6.220.2.3 struct timeval [ng_source_stats::lastTime](#)

Definition at line 63 of file `ng_source.h`.

6.220.2.4 uint32_t [ng_source_stats::maxPps](#)

Definition at line 59 of file `ng_source.h`.

6.220.2.5 uint64_t [ng_source_stats::outFrames](#)

Definition at line 56 of file `ng_source.h`.

6.220.2.6 uint64_t [ng_source_stats::outOctets](#)

Definition at line 55 of file `ng_source.h`.

6.220.2.7 `uint32_t` `ng_source_stats::queueFrames`

Definition at line 58 of file `ng_source.h`.

6.220.2.8 `uint32_t` `ng_source_stats::queueOctets`

Definition at line 57 of file `ng_source.h`.

6.220.2.9 `struct timeval` `ng_source_stats::startTime`

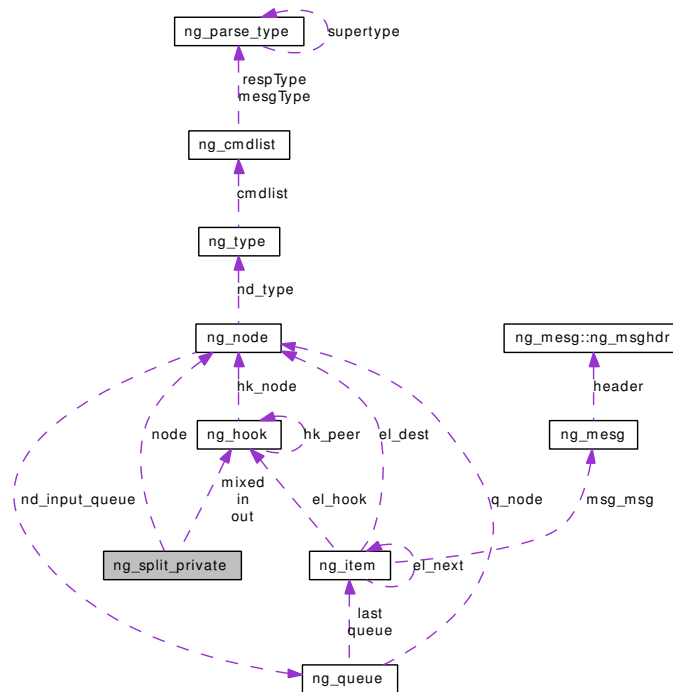
Definition at line 60 of file `ng_source.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_source.h](#)

6.221 ng_split_private Struct Reference

Collaboration diagram for ng_split_private:



Data Fields

- `hook_p out`
- `hook_p in`
- `hook_p mixed`
- `node_p node`

6.221.1 Detailed Description

Definition at line 67 of file `ng_split.c`.

6.221.2 Field Documentation

6.221.2.1 `hook_p ng_split_private::in`

Definition at line 69 of file `ng_split.c`.

6.221.2.2 `hook_p ng_split_private::mixed`

Definition at line 70 of file `ng_split.c`.

6.221.2.3 `node_p ng_split_private::node`

Definition at line 71 of file `ng_split.c`.

Referenced by `ng_split_constructor()`, `ng_split_newhook()`, and `ng_split_shutdown()`.

6.221.2.4 `hook_p ng_split_private::out`

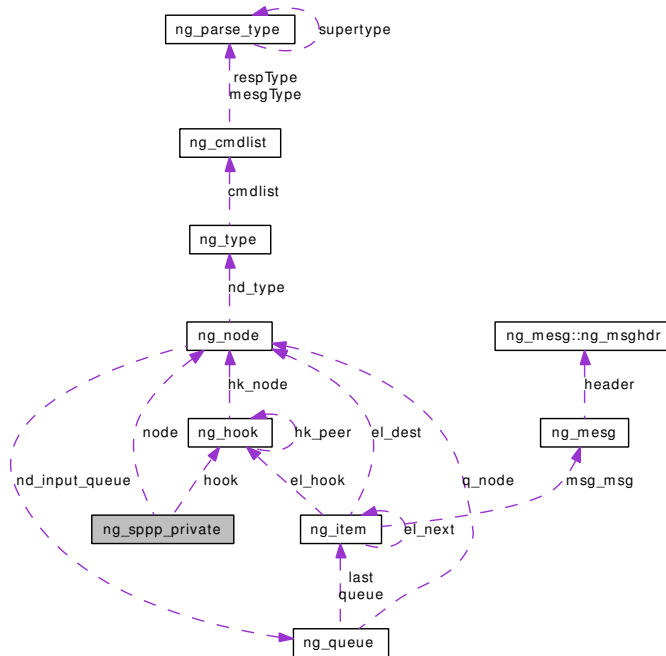
Definition at line 68 of file `ng_split.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_split.c](#)

6.222 ng_sppp_private Struct Reference

Collaboration diagram for ng_sppp_private:



Data Fields

- `ifnet *` `ifp`
- `int` `unit`
- `node_p` `node`
- `hook_p` `hook`

6.222.1 Detailed Description

Definition at line 53 of file `ng_sppp.c`.

6.222.2 Field Documentation

6.222.2.1 `hook_p` `ng_sppp_private::hook`

Definition at line 57 of file `ng_sppp.c`.

6.222.2.2 `struct ifnet*` `ng_sppp_private::ifp`

Definition at line 54 of file `ng_sppp.c`.

6.222.2.3 [node_p ng_sppp_private::node](#)

Definition at line 56 of file ng_sppp.c.

6.222.2.4 [int ng_sppp_private::unit](#)

Definition at line 55 of file ng_sppp.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_sppp.c](#)

6.223 ng_sscfu_getdefparam Struct Reference

```
#include <ng_sscfu.h>
```

Data Fields

- sscop_param [param](#)
- uint32_t [mask](#)

6.223.1 Detailed Description

Definition at line 50 of file [ng_sscfu.h](#).

6.223.2 Field Documentation

6.223.2.1 uint32_t [ng_sscfu_getdefparam::mask](#)

Definition at line 52 of file [ng_sscfu.h](#).

6.223.2.2 struct sscop_param [ng_sscfu_getdefparam::param](#)

Definition at line 51 of file [ng_sscfu.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_sscfu.h](#)

6.224 ng_sscop_setparam Struct Reference

```
#include <ng_sscop.h>
```

Data Fields

- [uint32_t mask](#)
- [sscop_param param](#)

6.224.1 Detailed Description

Definition at line 69 of file [ng_sscop.h](#).

6.224.2 Field Documentation

6.224.2.1 [uint32_t ng_sscop_setparam::mask](#)

Definition at line 70 of file [ng_sscop.h](#).

6.224.2.2 [struct sscop_param ng_sscop_setparam::param](#)

Definition at line 71 of file [ng_sscop.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_sscop.h](#)

6.225 ng_sscop_setparam_resp Struct Reference

```
#include <ng_sscop.h>
```

Data Fields

- [uint32_t mask](#)
- [int32_t error](#)

6.225.1 Detailed Description

Definition at line 80 of file [ng_sscop.h](#).

6.225.2 Field Documentation

6.225.2.1 [int32_t ng_sscop_setparam_resp::error](#)

Definition at line 82 of file [ng_sscop.h](#).

6.225.2.2 [uint32_t ng_sscop_setparam_resp::mask](#)

Definition at line 81 of file [ng_sscop.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_sscop.h](#)

6.226 ng_tag_hookin Struct Reference

```
#include <ng_tag.h>
```

Data Fields

- char [thisHook](#) [NG_HOOKSIZ]
- char [ifMatch](#) [NG_HOOKSIZ]
- char [ifNotMatch](#) [NG_HOOKSIZ]
- uint8_t [strip](#)
- uint32_t [tag_cookie](#)
- uint16_t [tag_id](#)
- uint16_t [tag_len](#)
- uint8_t [tag_data](#) [0]

6.226.1 Detailed Description

Definition at line 44 of file ng_tag.h.

6.226.2 Field Documentation

6.226.2.1 char [ng_tag_hookin::ifMatch](#)[NG_HOOKSIZ]

Definition at line 46 of file ng_tag.h.

6.226.2.2 char [ng_tag_hookin::ifNotMatch](#)[NG_HOOKSIZ]

Definition at line 47 of file ng_tag.h.

6.226.2.3 uint8_t [ng_tag_hookin::strip](#)

Definition at line 48 of file ng_tag.h.

6.226.2.4 uint32_t [ng_tag_hookin::tag_cookie](#)

Definition at line 49 of file ng_tag.h.

6.226.2.5 uint8_t [ng_tag_hookin::tag_data](#)[0]

Definition at line 52 of file ng_tag.h.

Referenced by [ng_tag_hookinary_getLength\(\)](#).

6.226.2.6 uint16_t [ng_tag_hookin::tag_id](#)

Definition at line 50 of file ng_tag.h.

6.226.2.7 `uint16_t ng_tag_hookin::tag_len`

Definition at line 51 of file `ng_tag.h`.

Referenced by `ng_tag_hookinary_getLength()`, `ng_tag_rcvmsg()`, and `ng_tag_setdata_in()`.

6.226.2.8 `char ng_tag_hookin::thisHook[NG_HOOKSIZ]`

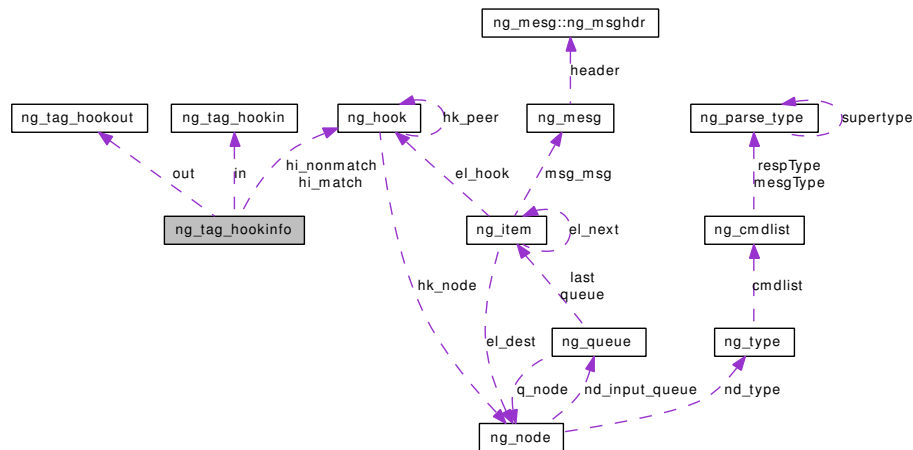
Definition at line 45 of file `ng_tag.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_tag.h`

6.227 ng_tag_hookinfo Struct Reference

Collaboration diagram for ng_tag_hookinfo:



Data Fields

- [hook_p hi_match](#)
- [hook_p hi_nonmatch](#)
- [uint32_t in_tag_cookie](#)
- [uint32_t out_tag_cookie](#)
- [uint16_t in_tag_id](#)
- [uint16_t in_tag_len](#)
- [uint16_t out_tag_id](#)
- [uint16_t out_tag_len](#)
- [uint8_t strip](#)
- [void * in_tag_data](#)
- [void * out_tag_data](#)
- [ng_tag_hookin * in](#)
- [ng_tag_hookout * out](#)

6.227.1 Detailed Description

Definition at line 94 of file ng_tag.c.

6.227.2 Field Documentation

6.227.2.1 [hook_p ng_tag_hookinfo::hi_match](#)

Definition at line 95 of file ng_tag.c.

6.227.2.2 [hook_p ng_tag_hookinfo::hi_nonmatch](#)

Definition at line 96 of file ng_tag.c.

6.227.2.3 struct [ng_tag_hookin*](#) [ng_tag_hookinfo::in](#)

Definition at line 106 of file [ng_tag.c](#).

6.227.2.4 uint32_t [ng_tag_hookinfo::in_tag_cookie](#)

Definition at line 97 of file [ng_tag.c](#).

6.227.2.5 void* [ng_tag_hookinfo::in_tag_data](#)

Definition at line 104 of file [ng_tag.c](#).

6.227.2.6 uint16_t [ng_tag_hookinfo::in_tag_id](#)

Definition at line 99 of file [ng_tag.c](#).

6.227.2.7 uint16_t [ng_tag_hookinfo::in_tag_len](#)

Definition at line 100 of file [ng_tag.c](#).

6.227.2.8 struct [ng_tag_hookout*](#) [ng_tag_hookinfo::out](#)

Definition at line 107 of file [ng_tag.c](#).

6.227.2.9 uint32_t [ng_tag_hookinfo::out_tag_cookie](#)

Definition at line 98 of file [ng_tag.c](#).

6.227.2.10 void* [ng_tag_hookinfo::out_tag_data](#)

Definition at line 105 of file [ng_tag.c](#).

6.227.2.11 uint16_t [ng_tag_hookinfo::out_tag_id](#)

Definition at line 101 of file [ng_tag.c](#).

6.227.2.12 uint16_t [ng_tag_hookinfo::out_tag_len](#)

Definition at line 102 of file [ng_tag.c](#).

6.227.2.13 uint8_t [ng_tag_hookinfo::strip](#)

Definition at line 103 of file [ng_tag.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tag.c](#)

6.228 ng_tag_hookout Struct Reference

```
#include <ng_tag.h>
```

Data Fields

- char [thisHook](#) [NG_HOOKSIZ]
- uint32_t [tag_cookie](#)
- uint16_t [tag_id](#)
- uint16_t [tag_len](#)
- uint8_t [tag_data](#) [0]

6.228.1 Detailed Description

Definition at line 56 of file [ng_tag.h](#).

6.228.2 Field Documentation

6.228.2.1 uint32_t [ng_tag_hookout::tag_cookie](#)

Definition at line 58 of file [ng_tag.h](#).

6.228.2.2 uint8_t [ng_tag_hookout::tag_data](#)[0]

Definition at line 61 of file [ng_tag.h](#).

Referenced by [ng_tag_hookoutary_getLength\(\)](#).

6.228.2.3 uint16_t [ng_tag_hookout::tag_id](#)

Definition at line 59 of file [ng_tag.h](#).

6.228.2.4 uint16_t [ng_tag_hookout::tag_len](#)

Definition at line 60 of file [ng_tag.h](#).

Referenced by [ng_tag_hookoutary_getLength\(\)](#), [ng_tag_rcvmsg\(\)](#), and [ng_tag_setdata_out\(\)](#).

6.228.2.5 char [ng_tag_hookout::thisHook](#)[NG_HOOKSIZ]

Definition at line 57 of file [ng_tag.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tag.h](#)

6.229 ng_tag_prio Struct Reference

```
#include <netgraph.h>
```

Data Fields

- m_tag [tag](#)
- char [priority](#)
- char [discardability](#)

6.229.1 Detailed Description

Definition at line 1120 of file netgraph.h.

6.229.2 Field Documentation

6.229.2.1 char [ng_tag_prio::discardability](#)

Definition at line 1123 of file netgraph.h.

Referenced by [nglmi_inquire\(\)](#).

6.229.2.2 char [ng_tag_prio::priority](#)

Definition at line 1122 of file netgraph.h.

Referenced by [nglmi_inquire\(\)](#).

6.229.2.3 struct m_tag [ng_tag_prio::tag](#)

Definition at line 1121 of file netgraph.h.

Referenced by [nglmi_inquire\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netgraph.h](#)

6.230 ng_tcpmss_config Struct Reference

```
#include <ng_tcpmss.h>
```

Data Fields

- char [inHook](#) [NG_HOOKSIZ]
- char [outHook](#) [NG_HOOKSIZ]
- uint16_t [maxMSS](#)

6.230.1 Detailed Description

Definition at line 60 of file [ng_tcpmss.h](#).

6.230.2 Field Documentation

6.230.2.1 char [ng_tcpmss_config::inHook](#)[NG_HOOKSIZ]

Definition at line 61 of file [ng_tcpmss.h](#).

Referenced by [ng_tcpmss_rcvmsg\(\)](#).

6.230.2.2 uint16_t [ng_tcpmss_config::maxMSS](#)

Definition at line 63 of file [ng_tcpmss.h](#).

Referenced by [ng_tcpmss_rcvmsg\(\)](#).

6.230.2.3 char [ng_tcpmss_config::outHook](#)[NG_HOOKSIZ]

Definition at line 62 of file [ng_tcpmss.h](#).

Referenced by [ng_tcpmss_rcvmsg\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tcpmss.h](#)

6.231 ng_tcpmss_hookstat Struct Reference

```
#include <ng_tcpmss.h>
```

Data Fields

- [uint64_t Octets](#)
- [uint64_t Packets](#)
- [uint16_t maxMSS](#)
- [uint64_t SYNPkts](#)
- [uint64_t FixedPkts](#)

6.231.1 Detailed Description

Definition at line 40 of file `ng_tcpmss.h`.

6.231.2 Field Documentation

6.231.2.1 [uint64_t ng_tcpmss_hookstat::FixedPkts](#)

Definition at line 45 of file `ng_tcpmss.h`.

Referenced by `ng_tcpmss_rcvdata()`.

6.231.2.2 [uint16_t ng_tcpmss_hookstat::maxMSS](#)

Definition at line 43 of file `ng_tcpmss.h`.

Referenced by `ng_tcpmss_rcvdata()`, and `ng_tcpmss_rcvmsg()`.

6.231.2.3 [uint64_t ng_tcpmss_hookstat::Octets](#)

Definition at line 41 of file `ng_tcpmss.h`.

Referenced by `ng_tcpmss_rcvdata()`.

6.231.2.4 [uint64_t ng_tcpmss_hookstat::Packets](#)

Definition at line 42 of file `ng_tcpmss.h`.

Referenced by `ng_tcpmss_rcvdata()`.

6.231.2.5 [uint64_t ng_tcpmss_hookstat::SYNPkts](#)

Definition at line 44 of file `ng_tcpmss.h`.

Referenced by `ng_tcpmss_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tcpmss.h](#)

6.232 ng_tee_hookstat Struct Reference

```
#include <ng_tee.h>
```

Data Fields

- [u_int64_t inOctets](#)
- [u_int64_t inFrames](#)
- [u_int64_t outOctets](#)
- [u_int64_t outFrames](#)

6.232.1 Detailed Description

Definition at line 59 of file `ng_tee.h`.

6.232.2 Field Documentation

6.232.2.1 [u_int64_t ng_tee_hookstat::inFrames](#)

Definition at line 61 of file `ng_tee.h`.

Referenced by `ngt_rcvdata()`.

6.232.2.2 [u_int64_t ng_tee_hookstat::inOctets](#)

Definition at line 60 of file `ng_tee.h`.

Referenced by `ngt_rcvdata()`.

6.232.2.3 [u_int64_t ng_tee_hookstat::outFrames](#)

Definition at line 63 of file `ng_tee.h`.

Referenced by `ngt_rcvdata()`.

6.232.2.4 [u_int64_t ng_tee_hookstat::outOctets](#)

Definition at line 62 of file `ng_tee.h`.

Referenced by `ngt_rcvdata()`.

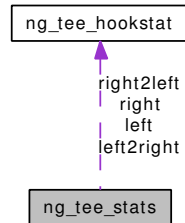
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tee.h](#)

6.233 ng_tee_stats Struct Reference

```
#include <ng_tee.h>
```

Collaboration diagram for ng_tee_stats:



Data Fields

- [ng_tee_hookstat right](#)
- [ng_tee_hookstat left](#)
- [ng_tee_hookstat right2left](#)
- [ng_tee_hookstat left2right](#)

6.233.1 Detailed Description

Definition at line 76 of file ng_tee.h.

6.233.2 Field Documentation

6.233.2.1 struct [ng_tee_hookstat](#) [ng_tee_stats::left](#)

Definition at line 78 of file ng_tee.h.

6.233.2.2 struct [ng_tee_hookstat](#) [ng_tee_stats::left2right](#)

Definition at line 80 of file ng_tee.h.

6.233.2.3 struct [ng_tee_hookstat](#) [ng_tee_stats::right](#)

Definition at line 77 of file ng_tee.h.

6.233.2.4 struct [ng_tee_hookstat](#) [ng_tee_stats::right2left](#)

Definition at line 79 of file ng_tee.h.

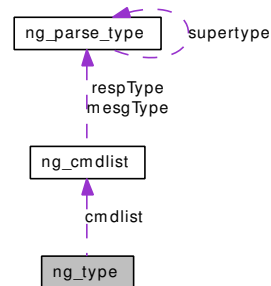
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_tee.h](#)

6.234 ng_type Struct Reference

```
#include <netgraph.h>
```

Collaboration diagram for ng_type:



Public Member Functions

- [LIST_ENTRY](#) ([ng_type](#)) types

Data Fields

- [u_int32_t](#) [version](#)
- [const char *](#) [name](#)
- [modeventhand_t](#) [mod_event](#)
- [ng_constructor_t *](#) [constructor](#)
- [ng_rcvmsg_t *](#) [rcvmsg](#)
- [ng_close_t *](#) [close](#)
- [ng_shutdown_t *](#) [shutdown](#)
- [ng_newhook_t *](#) [newhook](#)
- [ng_findhook_t *](#) [findhook](#)
- [ng_connect_t *](#) [connect](#)
- [ng_rcvdata_t *](#) [rcvdata](#)
- [ng_disconnect_t *](#) [disconnect](#)
- [ng_cmdlist *](#) [cmdlist](#)
- [int](#) [refs](#)

6.234.1 Detailed Description

Definition at line 1011 of file netgraph.h.

6.234.2 Member Function Documentation

6.234.2.1 `ng_type::LIST_ENTRY` (`ng_type`)

6.234.3 Field Documentation

6.234.3.1 `ng_close_t*` `ng_type::close`

Definition at line 1018 of file netgraph.h.

Referenced by `ng_rmnode()`.

6.234.3.2 `struct ng_cmdlist*` `ng_type::cmdlist`

Definition at line 1026 of file netgraph.h.

Referenced by `ng_generic_msg()`.

6.234.3.3 `ng_connect_t*` `ng_type::connect`

Definition at line 1022 of file netgraph.h.

Referenced by `ng_con_part2()`, `ng_con_part3()`, and `ng_mkpeer()`.

6.234.3.4 `ng_constructor_t*` `ng_type::constructor`

Definition at line 1016 of file netgraph.h.

Referenced by `ng_make_node()`.

6.234.3.5 `ng_disconnect_t*` `ng_type::disconnect`

Definition at line 1024 of file netgraph.h.

Referenced by `ng_destroy_hook()`.

6.234.3.6 `ng_findhook_t*` `ng_type::findhook`

Definition at line 1021 of file netgraph.h.

Referenced by `ng_findhook()`.

6.234.3.7 `modeventhand_t` `ng_type::mod_event`

Definition at line 1015 of file netgraph.h.

Referenced by `ng_mod_event()`.

6.234.3.8 `const char*` `ng_type::name`

Definition at line 1014 of file netgraph.h.

Referenced by `ng_connect_data()`, `ng_findtype()`, `ng_generic_msg()`, `ng_h4_ioctl()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_make_node_common()`, `ng_newtype()`, `ngc_send()`, `ngt_open()`, and `ngt_ioctl()`.

6.234.3.9 `ng_newhook_t*` `ng_type::newhook`

Definition at line 1020 of file `netgraph.h`.

Referenced by `ng_add_hook()`, and `ng_con_part2()`.

6.234.3.10 `ng_rcvdata_t*` `ng_type::rcvdata`

Definition at line 1023 of file `netgraph.h`.

6.234.3.11 `ng_rcvmsg_t*` `ng_type::rcvmsg`

Definition at line 1017 of file `netgraph.h`.

Referenced by `ng_apply_item()`.

6.234.3.12 `int` `ng_type::refs`

Definition at line 1030 of file `netgraph.h`.

Referenced by `ng_generic_msg()`, `ng_make_node_common()`, `ng_mod_event()`, `ng_newtype()`, `ng_rmtype()`, and `ng_unref_node()`.

6.234.3.13 `ng_shutdown_t*` `ng_type::shutdown`

Definition at line 1019 of file `netgraph.h`.

Referenced by `ng_rmnode()`.

6.234.3.14 `u_int32_t` `ng_type::version`

Definition at line 1013 of file `netgraph.h`.

Referenced by `ng_newtype()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/netgraph.h](#)

6.235 `ng_ubt_node_qlen_ep` Struct Reference

```
#include <ng_ubt.h>
```

Data Fields

- `int32_t queue`
- `int32_t qlen`

6.235.1 Detailed Description

Definition at line 66 of file `ng_ubt.h`.

6.235.2 Field Documentation

6.235.2.1 `int32_t ng_ubt_node_qlen_ep::qlen`

Definition at line 72 of file `ng_ubt.h`.

6.235.2.2 `int32_t ng_ubt_node_qlen_ep::queue`

Definition at line 67 of file `ng_ubt.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_ubt.h`

6.236 ng_ubt_node_stat_ep Struct Reference

```
#include <ng_ubt.h>
```

Data Fields

- [u_int32_t pckts_rcv](#)
- [u_int32_t bytes_rcv](#)
- [u_int32_t pckts_sent](#)
- [u_int32_t bytes_sent](#)
- [u_int32_t oerrors](#)
- [u_int32_t ierrors](#)

6.236.1 Detailed Description

Definition at line 76 of file `ng_ubt.h`.

6.236.2 Field Documentation

6.236.2.1 [u_int32_t ng_ubt_node_stat_ep::bytes_rcv](#)

Definition at line 78 of file `ng_ubt.h`.

6.236.2.2 [u_int32_t ng_ubt_node_stat_ep::bytes_sent](#)

Definition at line 80 of file `ng_ubt.h`.

6.236.2.3 [u_int32_t ng_ubt_node_stat_ep::ierrors](#)

Definition at line 82 of file `ng_ubt.h`.

6.236.2.4 [u_int32_t ng_ubt_node_stat_ep::oerrors](#)

Definition at line 81 of file `ng_ubt.h`.

6.236.2.5 [u_int32_t ng_ubt_node_stat_ep::pckts_rcv](#)

Definition at line 77 of file `ng_ubt.h`.

6.236.2.6 [u_int32_t ng_ubt_node_stat_ep::pckts_sent](#)

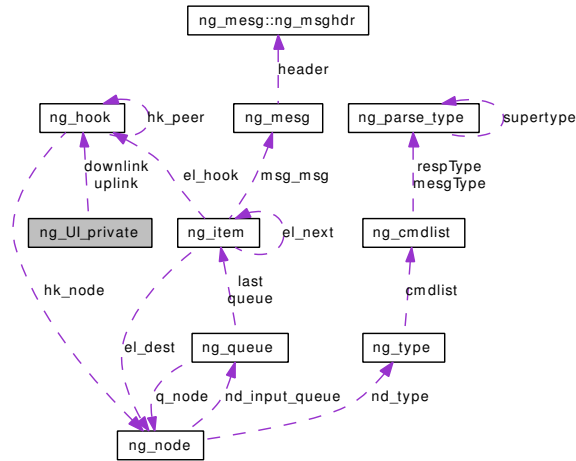
Definition at line 79 of file `ng_ubt.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_ubt.h](#)

6.237 ng_UI_private Struct Reference

Collaboration diagram for ng_UI_private:



Data Fields

- [hook_p downlink](#)
- [hook_p uplink](#)

6.237.1 Detailed Description

Definition at line 65 of file `ng_UI.c`.

6.237.2 Field Documentation

6.237.2.1 [hook_p ng_UI_private::downlink](#)

Definition at line 66 of file `ng_UI.c`.

6.237.2.2 [hook_p ng_UI_private::uplink](#)

Definition at line 67 of file `ng_UI.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_UI.c`

6.238 ng_vatmpif_config Struct Reference

```
#include <ng_atmpif.h>
```

Data Fields

- uint8_t [debug](#)
- uint32_t [pcr](#)
- Mac_addr [macaddr](#)

6.238.1 Detailed Description

Definition at line 53 of file ng_atmpif.h.

6.238.2 Field Documentation

6.238.2.1 uint8_t ng_vatmpif_config::debug

Definition at line 54 of file ng_atmpif.h.

6.238.2.2 Mac_addr ng_vatmpif_config::macaddr

Definition at line 56 of file ng_atmpif.h.

Referenced by vatmpif_harp_attach().

6.238.2.3 uint32_t ng_vatmpif_config::pcr

Definition at line 55 of file ng_atmpif.h.

Referenced by vatmpif_harp_attach().

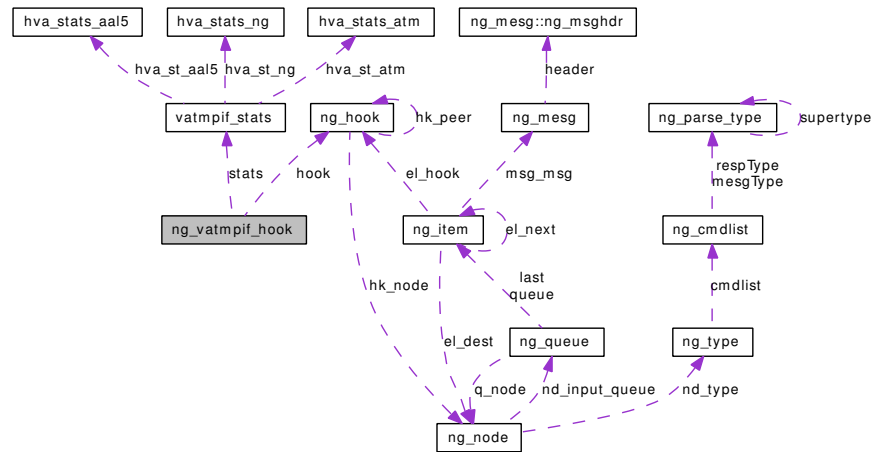
The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/atm/ng_atmpif.h

6.239 ng_vatmpif_hook Struct Reference

```
#include <ng_atmpif_var.h>
```

Collaboration diagram for ng_vatmpif_hook:



Data Fields

- [hook_p](#) hook
- [Vatmpif_stats](#) stats
- [uint32_t](#) InSeq
- [uint32_t](#) OutSeq
- [uint32_t](#) cur_pcr

6.239.1 Detailed Description

Definition at line 84 of file `ng_atmpif_var.h`.

6.239.2 Field Documentation

6.239.2.1 [uint32_t ng_vatmpif_hook::cur_pcr](#)

Definition at line 89 of file `ng_atmpif_var.h`.

Referenced by `ng_atmpif_rcvmsg()`.

6.239.2.2 [hook_p ng_vatmpif_hook::hook](#)

Definition at line 85 of file `ng_atmpif_var.h`.

Referenced by `ng_atmpif_disconnect()`, and `ng_atmpif_rcvdata()`.

6.239.2.3 `uint32_t ng_vatmpif_hook::InSeq`

Definition at line 87 of file `ng_atmpif_var.h`.

Referenced by `ng_atmpif_rcvmsg()`.

6.239.2.4 `uint32_t ng_vatmpif_hook::OutSeq`

Definition at line 88 of file `ng_atmpif_var.h`.

Referenced by `ng_atmpif_rcvmsg()`.

6.239.2.5 `Vatmpif_stats ng_vatmpif_hook::stats`

Definition at line 86 of file `ng_atmpif_var.h`.

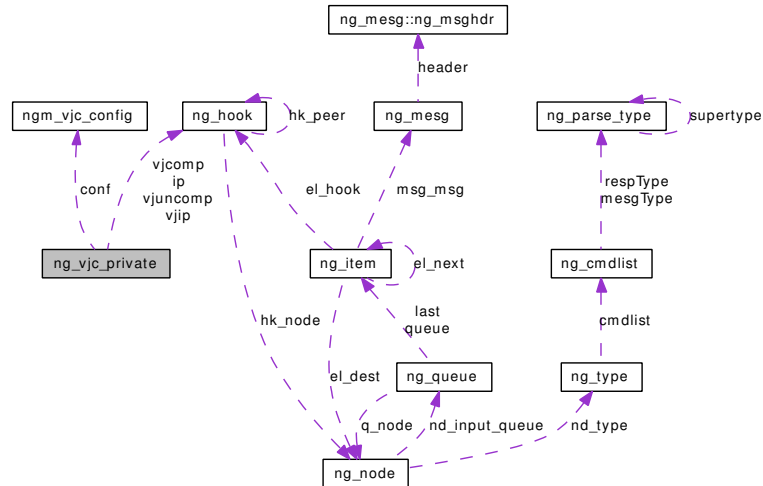
Referenced by `ng_atmpif_rcvmsg()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h](#)

6.240 ng_vjc_private Struct Reference

Collaboration diagram for ng_vjc_private:



Data Fields

- [ngm_vjc_config](#) `conf`
- `slcompress` `slc`
- `hook_p` `ip`
- `hook_p` `vjcomp`
- `hook_p` `vjuncomp`
- `hook_p` `vjip`

6.240.1 Detailed Description

Definition at line 79 of file `ng_vjc.c`.

6.240.2 Field Documentation

6.240.2.1 struct `ngm_vjc_config` `ng_vjc_private::conf`

Definition at line 80 of file `ng_vjc.c`.

6.240.2.2 `hook_p` `ng_vjc_private::ip`

Definition at line 82 of file `ng_vjc.c`.

6.240.2.3 struct `slcompress` `ng_vjc_private::slc`

Definition at line 81 of file `ng_vjc.c`.

6.240.2.4 [hook_p ng_vjc_private::vjcomp](#)

Definition at line 83 of file [ng_vjc.c](#).

6.240.2.5 [hook_p ng_vjc_private::vjip](#)

Definition at line 85 of file [ng_vjc.c](#).

6.240.2.6 [hook_p ng_vjc_private::vjuncomp](#)

Definition at line 84 of file [ng_vjc.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_vjc.c](#)

6.241 ng_vlan_filter Struct Reference

```
#include <ng_vlan.h>
```

Data Fields

- char [hook](#) [NG_HOOKSIZ]
- u_int16_t [vlan](#)

6.241.1 Detailed Description

Definition at line 50 of file [ng_vlan.h](#).

6.241.2 Field Documentation

6.241.2.1 char [ng_vlan_filter::hook](#)[NG_HOOKSIZ]

Definition at line 51 of file [ng_vlan.h](#).

Referenced by [ng_vlan_rcvmsg\(\)](#).

6.241.2.2 u_int16_t [ng_vlan_filter::vlan](#)

Definition at line 52 of file [ng_vlan.h](#).

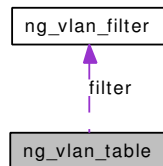
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_vlan.h](#)

6.242 ng_vlan_table Struct Reference

```
#include <ng_vlan.h>
```

Collaboration diagram for ng_vlan_table:



Data Fields

- `u_int32_t n`
- `ng_vlan_filter filter []`

6.242.1 Detailed Description

Definition at line 63 of file `ng_vlan.h`.

6.242.2 Field Documentation

6.242.2.1 struct `ng_vlan_filter ng_vlan_table::filter[]`

Definition at line 65 of file `ng_vlan.h`.

6.242.2.2 `u_int32_t ng_vlan_table::n`

Definition at line 64 of file `ng_vlan.h`.

Referenced by `ng_vlan_getTableLength()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_vlan.h`

6.243 ngatm_msg Struct Reference

6.243.1 Detailed Description

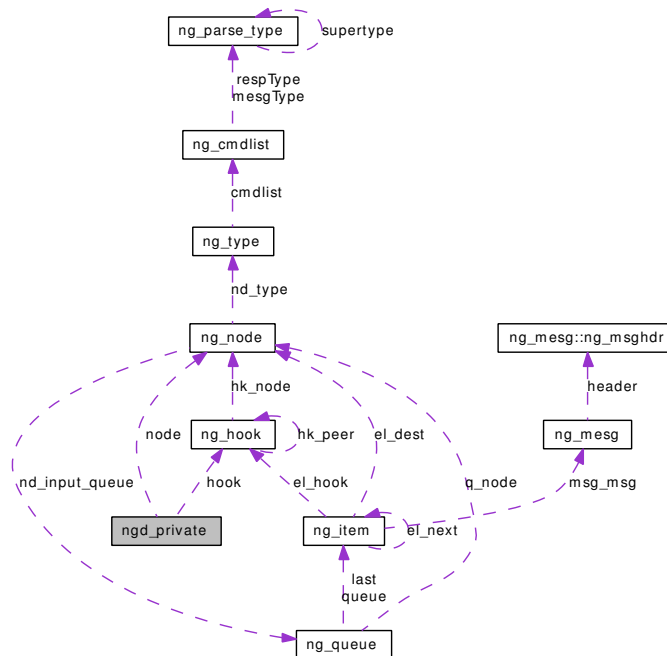
Definition at line 323 of file ngatmbase.c.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ngatmbase.c](#)

6.244 ngd_private Struct Reference

Collaboration diagram for ngd_private:



Data Fields

- ifqueue [readq](#)
- [ng_node](#) * [node](#)
- [ng_hook](#) * [hook](#)
- cdev * [ngddev](#)
- mtx [ngd_mtx](#)
- int [unit](#)
- [uint16_t](#) [flags](#)

6.244.1 Detailed Description

Definition at line 89 of file [ng_device.c](#).

6.244.2 Field Documentation

6.244.2.1 [uint16_t](#) [ngd_private::flags](#)

Definition at line 96 of file [ng_device.c](#).

6.244.2.2 struct [ng_hook](#)* [ngd_private::hook](#)

Definition at line 92 of file [ng_device.c](#).

6.244.2.3 struct mtx [ngd_private::ngd_mtx](#)

Definition at line 94 of file [ng_device.c](#).

6.244.2.4 struct cdev* [ngd_private::ngddev](#)

Definition at line 93 of file [ng_device.c](#).

6.244.2.5 struct [ng_node*](#) [ngd_private::node](#)

Definition at line 91 of file [ng_device.c](#).

6.244.2.6 struct ifqueue [ngd_private::readq](#)

Definition at line 90 of file [ng_device.c](#).

6.244.2.7 int [ngd_private::unit](#)

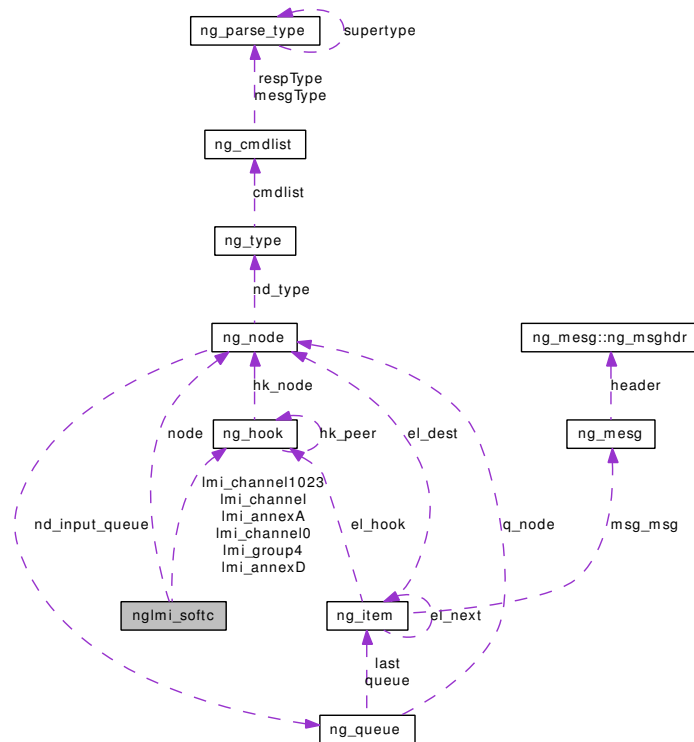
Definition at line 95 of file [ng_device.c](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_device.c](#)

6.245 nglmi_softc Struct Reference

Collaboration diagram for nglmi_softc:



Data Fields

- `node_p` `node`
- `int` `flags`
- `int` `poll_count`
- `int` `poll_state`
- `u_char` `remote_seq`
- `u_char` `local_seq`
- `u_char` `protoID`
- `u_long` `seq_retries`
- `callout` `handle`
- `int` `liv_per_full`
- `int` `liv_rate`
- `int` `livs`
- `int` `need_full`
- `hook_p` `lmi_channel`
- `hook_p` `lmi_annexA`
- `hook_p` `lmi_annexD`
- `hook_p` `lmi_group4`
- `hook_p` `lmi_channel0`
- `hook_p` `lmi_channel1023`
- `char *` `protoname`

- u_char [dlci_state](#) [MAXDLCI+1]
- int [invalidx](#)

6.245.1 Detailed Description

Definition at line 116 of file ng_lmi.c.

6.245.2 Field Documentation

6.245.2.1 u_char [nglmi_softc::dlci_state](#)[MAXDLCI+1]

Definition at line 137 of file ng_lmi.c.

6.245.2.2 int [nglmi_softc::flags](#)

Definition at line 118 of file ng_lmi.c.

6.245.2.3 struct callout [nglmi_softc::handle](#)

Definition at line 125 of file ng_lmi.c.

6.245.2.4 int [nglmi_softc::invalidx](#)

Definition at line 138 of file ng_lmi.c.

6.245.2.5 int [nglmi_softc::liv_per_full](#)

Definition at line 126 of file ng_lmi.c.

6.245.2.6 int [nglmi_softc::liv_rate](#)

Definition at line 127 of file ng_lmi.c.

6.245.2.7 int [nglmi_softc::lives](#)

Definition at line 128 of file ng_lmi.c.

6.245.2.8 hook_p [nglmi_softc::lmi_annexA](#)

Definition at line 131 of file ng_lmi.c.

6.245.2.9 hook_p [nglmi_softc::lmi_annexD](#)

Definition at line 132 of file ng_lmi.c.

6.245.2.10 hook_p nglmi_softc::lmi_channel

Definition at line 130 of file ng_lmi.c.

6.245.2.11 hook_p nglmi_softc::lmi_channel0

Definition at line 134 of file ng_lmi.c.

6.245.2.12 hook_p nglmi_softc::lmi_channel1023

Definition at line 135 of file ng_lmi.c.

6.245.2.13 hook_p nglmi_softc::lmi_group4

Definition at line 133 of file ng_lmi.c.

6.245.2.14 u_char nglmi_softc::local_seq

Definition at line 122 of file ng_lmi.c.

6.245.2.15 int nglmi_softc::need_full

Definition at line 129 of file ng_lmi.c.

6.245.2.16 node_p nglmi_softc::node

Definition at line 117 of file ng_lmi.c.

Referenced by LMI_ticker(), nglmi_constructor(), and nglmi_newhook().

6.245.2.17 int nglmi_softc::poll_count

Definition at line 119 of file ng_lmi.c.

6.245.2.18 int nglmi_softc::poll_state

Definition at line 120 of file ng_lmi.c.

6.245.2.19 u_char nglmi_softc::protoID

Definition at line 123 of file ng_lmi.c.

6.245.2.20 char* nglmi_softc::protoname

Definition at line 136 of file ng_lmi.c.

6.245.2.21 `u_char nglmi_softc::remote_seq`

Definition at line 121 of file `ng_lmi.c`.

6.245.2.22 `u_long nglmi_softc::seq_retries`

Definition at line 124 of file `ng_lmi.c`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_lmi.c](#)

6.246 nglmistat Struct Reference

```
#include <ng_lmi.h>
```

Data Fields

- u_char [proto](#) [12]
- u_char [hook](#) [12]
- u_char [fixed](#)
- u_char [autod](#)
- u_char [seen](#) [NGM_LMI_STAT_ARYSIZE]
- u_char [up](#) [NGM_LMI_STAT_ARYSIZE]

6.246.1 Detailed Description

Definition at line 66 of file `ng_lmi.h`.

6.246.2 Field Documentation

6.246.2.1 u_char [nglmistat::autod](#)

Definition at line 70 of file `ng_lmi.h`.

6.246.2.2 u_char [nglmistat::fixed](#)

Definition at line 69 of file `ng_lmi.h`.

6.246.2.3 u_char [nglmistat::hook](#)[12]

Definition at line 68 of file `ng_lmi.h`.

Referenced by `nglmi_rcvdata()`.

6.246.2.4 u_char [nglmistat::proto](#)[12]

Definition at line 67 of file `ng_lmi.h`.

6.246.2.5 u_char [nglmistat::seen](#)[NGM_LMI_STAT_ARYSIZE]

Definition at line 71 of file `ng_lmi.h`.

6.246.2.6 u_char [nglmistat::up](#)[NGM_LMI_STAT_ARYSIZE]

Definition at line 72 of file `ng_lmi.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_lmi.h`

6.247 ngm_atm_acr_change Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- [uint32_t node](#)
- [uint16_t vci](#)
- [uint8_t vpi](#)
- [uint32_t acr](#)

6.247.1 Detailed Description

Definition at line 233 of file `ng_atm.h`.

6.247.2 Field Documentation

6.247.2.1 [uint32_t ngm_atm_acr_change::acr](#)

Definition at line 237 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.247.2.2 [uint32_t ngm_atm_acr_change::node](#)

Definition at line 234 of file `ng_atm.h`.

Referenced by `ng_atm_event()`, and `ng_atm_event_func()`.

6.247.2.3 [uint16_t ngm_atm_acr_change::vci](#)

Definition at line 235 of file `ng_atm.h`.

6.247.2.4 [uint8_t ngm_atm_acr_change::vpi](#)

Definition at line 236 of file `ng_atm.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atm.h`

6.248 ngm_atm_config Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- [uint32_t pcr](#)
- [uint32_t vpi_bits](#)
- [uint32_t vci_bits](#)
- [uint32_t max_vpcs](#)
- [uint32_t max_vccs](#)

6.248.1 Detailed Description

Definition at line 62 of file `ng_atm.h`.

6.248.2 Field Documentation

6.248.2.1 [uint32_t ngm_atm_config::max_vccs](#)

Definition at line 67 of file `ng_atm.h`.

6.248.2.2 [uint32_t ngm_atm_config::max_vpcs](#)

Definition at line 66 of file `ng_atm.h`.

6.248.2.3 [uint32_t ngm_atm_config::pcr](#)

Definition at line 63 of file `ng_atm.h`.

6.248.2.4 [uint32_t ngm_atm_config::vci_bits](#)

Definition at line 65 of file `ng_atm.h`.

6.248.2.5 [uint32_t ngm_atm_config::vpi_bits](#)

Definition at line 64 of file `ng_atm.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atm.h`

6.249 ngm_atm_cpcs_init Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- char [name](#) [NG_HOOKSIZ]
- uint32_t [flags](#)
- uint16_t [vci](#)
- uint16_t [vpi](#)
- uint16_t [rmtu](#)
- uint16_t [tmtu](#)
- uint8_t [aal](#)
- uint8_t [traffic](#)
- uint32_t [pcr](#)
- uint32_t [scr](#)
- uint32_t [mbs](#)
- uint32_t [mcr](#)
- uint32_t [icr](#)
- uint32_t [tbe](#)
- uint8_t [nrm](#)
- uint8_t [trm](#)
- uint16_t [adtf](#)
- uint8_t [rif](#)
- uint8_t [rdf](#)
- uint8_t [cdf](#)

6.249.1 Detailed Description

Definition at line 130 of file `ng_atm.h`.

6.249.2 Field Documentation

6.249.2.1 uint8_t [ngm_atm_cpcs_init::aal](#)

Definition at line 137 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.2 uint16_t [ngm_atm_cpcs_init::adtf](#)

Definition at line 147 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.3 uint8_t [ngm_atm_cpcs_init::cdf](#)

Definition at line 150 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.4 `uint32_t ngm_atm_cpcs_init::flags`

Definition at line 132 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.5 `uint32_t ngm_atm_cpcs_init::icr`

Definition at line 143 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.6 `uint32_t ngm_atm_cpcs_init::mbs`

Definition at line 141 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.7 `uint32_t ngm_atm_cpcs_init::mcr`

Definition at line 142 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.8 `char ngm_atm_cpcs_init::name[NG_HOOKSIZ]`

Definition at line 131 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.9 `uint8_t ngm_atm_cpcs_init::nrm`

Definition at line 145 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.10 `uint32_t ngm_atm_cpcs_init::pcr`

Definition at line 139 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.11 `uint8_t ngm_atm_cpcs_init::rdf`

Definition at line 149 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.12 `uint8_t ngm_atm_cpcs_init::rif`

Definition at line 148 of file ng_atm.h.

Referenced by ng_atm_cpcs_init().

6.249.2.13 `uint16_t ngm_atm_cpcs_init::rmtu`

Definition at line 135 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.14 `uint32_t ngm_atm_cpcs_init::scr`

Definition at line 140 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.15 `uint32_t ngm_atm_cpcs_init::tbe`

Definition at line 144 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.16 `uint16_t ngm_atm_cpcs_init::tmtu`

Definition at line 136 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.17 `uint8_t ngm_atm_cpcs_init::traffic`

Definition at line 138 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.18 `uint8_t ngm_atm_cpcs_init::trm`

Definition at line 146 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.19 `uint16_t ngm_atm_cpcs_init::vci`

Definition at line 133 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

6.249.2.20 `uint16_t ngm_atm_cpcs_init::vpi`

Definition at line 134 of file `ng_atm.h`.

Referenced by `ng_atm_cpcs_init()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.h](#)

6.250 ngm_atm_cpcs_term Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- char [name](#) [NG_HOOKSIZ]

6.250.1 Detailed Description

Definition at line 181 of file [ng_atm.h](#).

6.250.2 Field Documentation

6.250.2.1 char [ngm_atm_cpcs_term::name](#)[NG_HOOKSIZ]

Definition at line 182 of file [ng_atm.h](#).

Referenced by [ng_atm_cpcs_term\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.h](#)

6.251 ngm_atm_if_change Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- [uint32_t node](#)
- [uint8_t carrier](#)
- [uint8_t running](#)

6.251.1 Detailed Description

Definition at line 205 of file `ng_atm.h`.

6.251.2 Field Documentation

6.251.2.1 [uint8_t ngm_atm_if_change::carrier](#)

Definition at line 207 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.251.2.2 [uint32_t ngm_atm_if_change::node](#)

Definition at line 206 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.251.2.3 [uint8_t ngm_atm_if_change::running](#)

Definition at line 208 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.h](#)

6.252 ngm_atm_stats Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- [uint64_t in_packets](#)
- [uint64_t in_errors](#)
- [uint64_t out_packets](#)
- [uint64_t out_errors](#)

6.252.1 Detailed Description

Definition at line 190 of file ng_atm.h.

6.252.2 Field Documentation

6.252.2.1 [uint64_t ngm_atm_stats::in_errors](#)

Definition at line 192 of file ng_atm.h.

6.252.2.2 [uint64_t ngm_atm_stats::in_packets](#)

Definition at line 191 of file ng_atm.h.

6.252.2.3 [uint64_t ngm_atm_stats::out_errors](#)

Definition at line 194 of file ng_atm.h.

6.252.2.4 [uint64_t ngm_atm_stats::out_packets](#)

Definition at line 193 of file ng_atm.h.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.h](#)

6.253 ngm_atm_vcc_change Struct Reference

```
#include <ng_atm.h>
```

Data Fields

- [uint32_t node](#)
- [uint16_t vci](#)
- [uint8_t vpi](#)
- [uint8_t state](#)

6.253.1 Detailed Description

Definition at line 218 of file `ng_atm.h`.

6.253.2 Field Documentation

6.253.2.1 [uint32_t ngm_atm_vcc_change::node](#)

Definition at line 219 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.253.2.2 [uint8_t ngm_atm_vcc_change::state](#)

Definition at line 222 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.253.2.3 [uint16_t ngm_atm_vcc_change::vci](#)

Definition at line 220 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

6.253.2.4 [uint8_t ngm_atm_vcc_change::vpi](#)

Definition at line 221 of file `ng_atm.h`.

Referenced by `ng_atm_event_func()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.h](#)

6.254 ngm_bandwidth Struct Reference

```
#include <ng_message.h>
```

Data Fields

- `u_int64_t` [nominal_in](#)
- `u_int64_t` [seen_in](#)
- `u_int64_t` [nominal_out](#)
- `u_int64_t` [seen_out](#)

6.254.1 Detailed Description

Definition at line 314 of file `ng_message.h`.

6.254.2 Field Documentation

6.254.2.1 `u_int64_t` [ngm_bandwidth::nominal_in](#)

Definition at line 315 of file `ng_message.h`.

6.254.2.2 `u_int64_t` [ngm_bandwidth::nominal_out](#)

Definition at line 317 of file `ng_message.h`.

6.254.2.3 `u_int64_t` [ngm_bandwidth::seen_in](#)

Definition at line 316 of file `ng_message.h`.

6.254.2.4 `u_int64_t` [ngm_bandwidth::seen_out](#)

Definition at line 318 of file `ng_message.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.255 ngm_ccatm_addr_req Struct Reference

```
#include <ng_ccatm.h>
```

Data Fields

- [uint32_t port](#)
- [uni_addr addr](#)

6.255.1 Detailed Description

Definition at line 83 of file `ng_ccatm.h`.

6.255.2 Field Documentation

6.255.2.1 `struct uni_addr ngm_ccatm_addr_req::addr`

Definition at line 85 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_get_addresses()`.

6.255.2.2 `uint32_t ngm_ccatm_addr_req::port`

Definition at line 84 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_get_addresses()`.

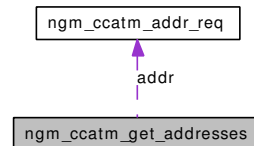
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_ccatm.h`

6.256 ngm_ccatm_get_addresses Struct Reference

```
#include <ng_ccatm.h>
```

Collaboration diagram for ngm_ccatm_get_addresses:



Data Fields

- `uint32_t count`
- `ngm_ccatm_addr_req addr []`

6.256.1 Detailed Description

Definition at line 97 of file `ng_ccatm.h`.

6.256.2 Field Documentation

6.256.2.1 struct `ngm_ccatm_addr_req ngm_ccatm_get_addresses::addr[]`

Definition at line 99 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_addr_req_array_getlen()`, and `ng_ccatm_get_addresses()`.

6.256.2.2 `uint32_t ngm_ccatm_get_addresses::count`

Definition at line 98 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_addr_req_array_getlen()`, and `ng_ccatm_get_addresses()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_ccatm.h`

6.257 ngm_ccatm_port Struct Reference

```
#include <ng_ccatm.h>
```

Data Fields

- [uint32_t port](#)

6.257.1 Detailed Description

Definition at line 116 of file `ng_ccatm.h`.

6.257.2 Field Documentation

6.257.2.1 `uint32_t ngm_ccatm_port::port`

Definition at line 117 of file `ng_ccatm.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_ccatm.h](#)

6.258 ngm_ccatm_portlist Struct Reference

```
#include <ng_ccatm.h>
```

Data Fields

- [uint32_t nports](#)
- [uint32_t ports \[\]](#)

6.258.1 Detailed Description

Definition at line 151 of file `ng_ccatm.h`.

6.258.2 Field Documentation

6.258.2.1 [uint32_t ngm_ccatm_portlist::nports](#)

Definition at line 152 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_port_array_getlen()`.

6.258.2.2 [uint32_t ngm_ccatm_portlist::ports\[\]](#)

Definition at line 153 of file `ng_ccatm.h`.

Referenced by `ng_ccatm_port_array_getlen()`, and `ng_ccatm_rcvmsg()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_ccatm.h`

6.259 ngm_connect Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [path](#) [NG_PATHSIZ]
- char [ourhook](#) [NG_HOOKSIZ]
- char [peerhook](#) [NG_HOOKSIZ]

6.259.1 Detailed Description

Definition at line 196 of file `ng_message.h`.

6.259.2 Field Documentation

6.259.2.1 char [ngm_connect::ourhook](#)[NG_HOOKSIZ]

Definition at line 198 of file `ng_message.h`.

6.259.2.2 char [ngm_connect::path](#)[NG_PATHSIZ]

Definition at line 197 of file `ng_message.h`.

6.259.2.3 char [ngm_connect::peerhook](#)[NG_HOOKSIZ]

Definition at line 199 of file `ng_message.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.260 ngm_mkpeer Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [type](#) [NG_TYPESIZ]
- char [ourhook](#) [NG_HOOKSIZ]
- char [peerhook](#) [NG_HOOKSIZ]

6.260.1 Detailed Description

Definition at line 181 of file [ng_message.h](#).

6.260.2 Field Documentation

6.260.2.1 char [ngm_mkpeer::ourhook](#)[NG_HOOKSIZ]

Definition at line 183 of file [ng_message.h](#).

6.260.2.2 char [ngm_mkpeer::peerhook](#)[NG_HOOKSIZ]

Definition at line 184 of file [ng_message.h](#).

6.260.2.3 char [ngm_mkpeer::type](#)[NG_TYPESIZ]

Definition at line 182 of file [ng_message.h](#).

Referenced by [ngc_send\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.261 ngm_name Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [name](#) [NG_NODESIZ]

6.261.1 Detailed Description

Definition at line 211 of file [ng_message.h](#).

6.261.2 Field Documentation

6.261.2.1 char [ngm_name::name](#)[NG_NODESIZ]

Definition at line 212 of file [ng_message.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.262 ngm_queue_state Struct Reference

```
#include <ng_message.h>
```

Data Fields

- [u_int queue_priority](#)
- [u_int max_queuelen_bytes](#)
- [u_int max_queuelen_packets](#)
- [u_int low_watermark](#)
- [u_int high_watermark](#)
- [u_int current](#)

6.262.1 Detailed Description

Definition at line 340 of file ng_message.h.

6.262.2 Field Documentation

6.262.2.1 [u_int ngm_queue_state::current](#)

Definition at line 346 of file ng_message.h.

Referenced by [flow_upper\(\)](#).

6.262.2.2 [u_int ngm_queue_state::high_watermark](#)

Definition at line 345 of file ng_message.h.

Referenced by [flow_upper\(\)](#).

6.262.2.3 [u_int ngm_queue_state::low_watermark](#)

Definition at line 344 of file ng_message.h.

6.262.2.4 [u_int ngm_queue_state::max_queuelen_bytes](#)

Definition at line 342 of file ng_message.h.

6.262.2.5 [u_int ngm_queue_state::max_queuelen_packets](#)

Definition at line 343 of file ng_message.h.

Referenced by [flow_upper\(\)](#).

6.262.2.6 `u_int ngm_queue_state::queue_priority`

Definition at line 341 of file `ng_message.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.263 ngm_rmhook Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [ourhook](#) [NG_HOOKSIZ]

6.263.1 Detailed Description

Definition at line 222 of file [ng_message.h](#).

6.263.2 Field Documentation

6.263.2.1 char [ngm_rmhook::ourhook](#)[NG_HOOKSIZ]

Definition at line 223 of file [ng_message.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.264 ngm_uni_config_mask Struct Reference

```
#include <ng_uni.h>
```

Data Fields

- [uint32_t mask](#)
- [uint32_t popt_mask](#)
- [uint32_t option_mask](#)

6.264.1 Detailed Description

Definition at line 86 of file `ng_uni.h`.

6.264.2 Field Documentation

6.264.2.1 [uint32_t ngm_uni_config_mask::mask](#)

Definition at line 87 of file `ng_uni.h`.

Referenced by `ng_uni_rcvmsg()`.

6.264.2.2 [uint32_t ngm_uni_config_mask::option_mask](#)

Definition at line 89 of file `ng_uni.h`.

6.264.2.3 [uint32_t ngm_uni_config_mask::popt_mask](#)

Definition at line 88 of file `ng_uni.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_uni.h](#)

6.265 ngm_uni_debug Struct Reference

```
#include <ng_uni.h>
```

Data Fields

- uint32_t [level](#) [UNI_MAXFACILITY]

6.265.1 Detailed Description

Definition at line 49 of file ng_uni.h.

6.265.2 Field Documentation

6.265.2.1 uint32_t [ngm_uni_debug::level](#)[UNI_MAXFACILITY]

Definition at line 50 of file ng_uni.h.

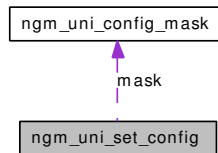
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_uni.h](#)

6.266 ngm_uni_set_config Struct Reference

```
#include <ng_uni.h>
```

Collaboration diagram for ngm_uni_set_config:



Data Fields

- uni_config [config](#)
- ngm_uni_config_mask [mask](#)

6.266.1 Detailed Description

Definition at line 99 of file ng_uni.h.

6.266.2 Field Documentation

6.266.2.1 struct uni_config [ngm_uni_set_config::config](#)

Definition at line 100 of file ng_uni.h.

6.266.2.2 struct [ngm_uni_config_mask](#) [ngm_uni_set_config::mask](#)

Definition at line 101 of file ng_uni.h.

Referenced by [ng_uni_rcvmsg\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_uni.h](#)

6.267 ngm_vjc_config Struct Reference

```
#include <ng_vjc.h>
```

Data Fields

- u_char [enableComp](#)
- u_char [enableDecomp](#)
- u_char [maxChannel](#)
- u_char [compressCID](#)

6.267.1 Detailed Description

Definition at line 63 of file `ng_vjc.h`.

6.267.2 Field Documentation

6.267.2.1 u_char [ngm_vjc_config::compressCID](#)

Definition at line 67 of file `ng_vjc.h`.

6.267.2.2 u_char [ngm_vjc_config::enableComp](#)

Definition at line 64 of file `ng_vjc.h`.

6.267.2.3 u_char [ngm_vjc_config::enableDecomp](#)

Definition at line 65 of file `ng_vjc.h`.

6.267.2.4 u_char [ngm_vjc_config::maxChannel](#)

Definition at line 66 of file `ng_vjc.h`.

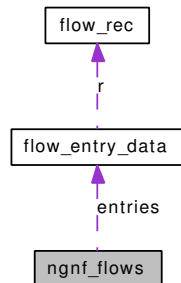
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_vjc.h`

6.268 ngnf_flows Struct Reference

```
#include <ng_netflow.h>
```

Collaboration diagram for ngnf_flows:



Data Fields

- `uint32_t nentries`
- `uint32_t last`
- `flow_entry_data entries [0]`

6.268.1 Detailed Description

Definition at line 149 of file `ng_netflow.h`.

6.268.2 Field Documentation

6.268.2.1 struct `flow_entry_data ngnf_flows::entries[0]`

Definition at line 152 of file `ng_netflow.h`.

Referenced by `ng_netflow_flow_show()`.

6.268.2.2 `uint32_t ngnf_flows::last`

Definition at line 151 of file `ng_netflow.h`.

Referenced by `ng_netflow_flow_show()`.

6.268.2.3 `uint32_t ngnf_flows::nentries`

Definition at line 150 of file `ng_netflow.h`.

Referenced by `ng_netflow_flow_show()`.

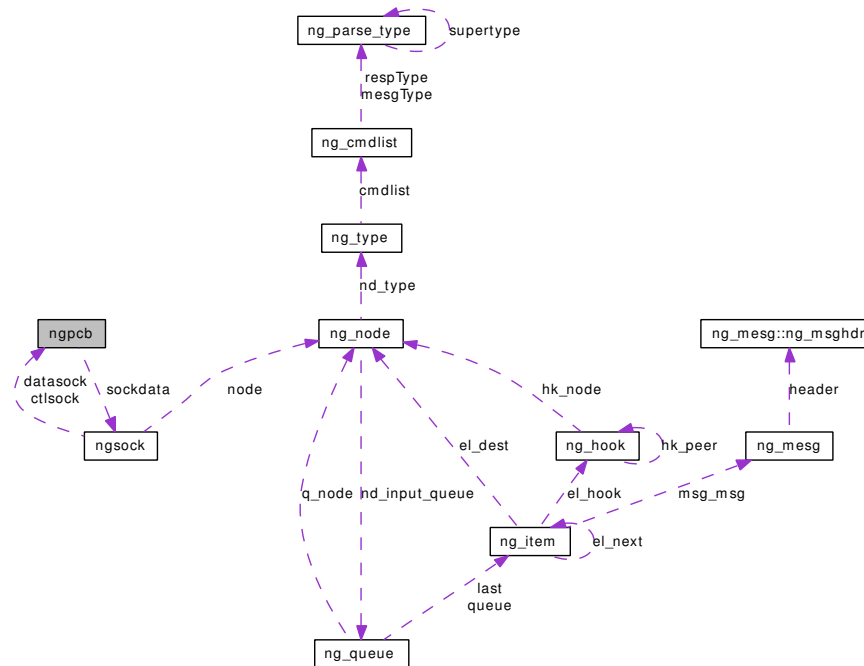
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/netflow/ng_netflow.h`

6.269 ngpcb Struct Reference

```
#include <ng_socketvar.h>
```

Collaboration diagram for ngpcb:



Public Member Functions

- [LIST_ENTRY](#) ([ngpcb](#)) socks

Data Fields

- socket * [ng_socket](#)
- [ngsock](#) * [sockdata](#)
- int [type](#)

6.269.1 Detailed Description

Definition at line 48 of file [ng_socketvar.h](#).

6.269.2 Member Function Documentation

6.269.2.1 `ngpcb::LIST_ENTRY` ([ngpcb](#))

6.269.3 Field Documentation

6.269.3.1 `struct socket*` [ngpcb::ng_socket](#)

Definition at line 49 of file `ng_socketvar.h`.

Referenced by `ng_attach_common()`, `ng_detach_common()`, `ngs_connect()`, `ngs_disconnect()`, `ngs_rcvdata()`, `ngs_rcvmsg()`, and `ngs_shutdown()`.

6.269.3.2 `struct ngsock*` [ngpcb::sockdata](#)

Definition at line 50 of file `ng_socketvar.h`.

Referenced by `ng_attach_cntl()`, `ng_bind()`, `ng_connect_data()`, `ng_detach_common()`, `ngc_send()`, and `ngd_send()`.

6.269.3.3 `int` [ngpcb::type](#)

Definition at line 52 of file `ng_socketvar.h`.

Referenced by `ng_attach_common()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_socketvar.h`

6.270 ngpppoe_init_data Struct Reference

```
#include <ng_pppoe.h>
```

Data Fields

- char [hook](#) [NG_HOOKSIZ]
- u_int16_t [data_len](#)
- char [data](#) []

6.270.1 Detailed Description

Definition at line 127 of file [ng_pppoe.h](#).

6.270.2 Field Documentation

6.270.2.1 char [ngpppoe_init_data::data](#) []

Definition at line 130 of file [ng_pppoe.h](#).

Referenced by [ng_pppoe_rcvmsg\(\)](#).

6.270.2.2 u_int16_t [ngpppoe_init_data::data_len](#)

Definition at line 129 of file [ng_pppoe.h](#).

Referenced by [ng_pppoe_rcvmsg\(\)](#).

6.270.2.3 char [ngpppoe_init_data::hook](#)[NG_HOOKSIZ]

Definition at line 128 of file [ng_pppoe.h](#).

Referenced by [ng_pppoe_rcvmsg\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.271 ngpppoe_sts Struct Reference

```
#include <ng_pppoe.h>
```

Data Fields

- char [hook](#) [NG_HOOKSIZ]

6.271.1 Detailed Description

Definition at line 145 of file ng_pppoe.h.

6.271.2 Field Documentation

6.271.2.1 char [ngpppoe_sts::hook](#)[NG_HOOKSIZ]

Definition at line 146 of file ng_pppoe.h.

Referenced by [pppoe_send_event\(\)](#), and [send_acname\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.272 ngpppoestat Struct Reference

```
#include <ng_pppoe.h>
```

Data Fields

- [u_int packets_in](#)
- [u_int packets_out](#)

6.272.1 Detailed Description

Definition at line 97 of file [ng_pppoe.h](#).

6.272.2 Field Documentation

6.272.2.1 [u_int ngpppoestat::packets_in](#)

Definition at line 98 of file [ng_pppoe.h](#).

6.272.2.2 [u_int ngpppoestat::packets_out](#)

Definition at line 99 of file [ng_pppoe.h](#).

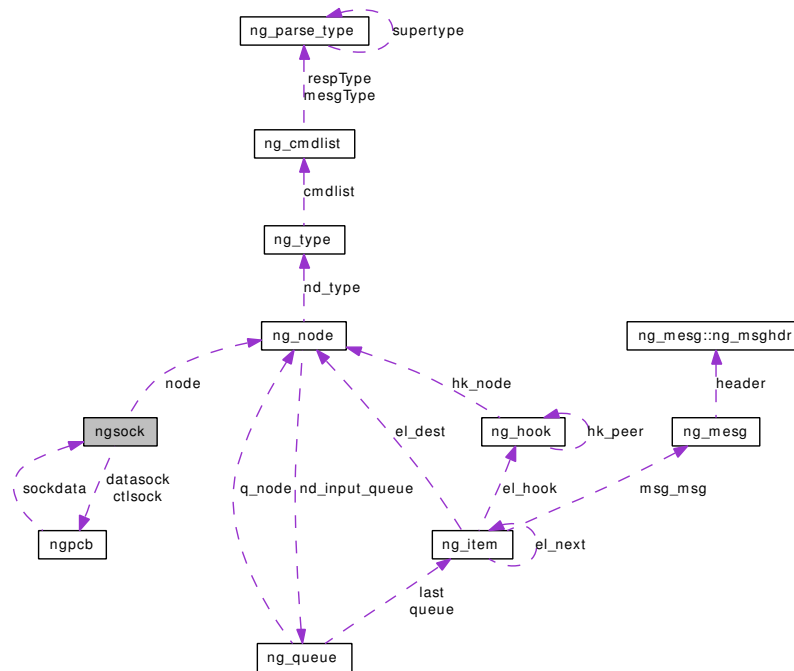
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.273 ngsock Struct Reference

```
#include <ng_socketvar.h>
```

Collaboration diagram for ngsock:



Data Fields

- `ng_node * node`
- `ngpcb * datasock`
- `ngpcb * ctlsock`
- `int flags`
- `int refs`
- `mtx mtx`
- `int error`

6.273.1 Detailed Description

Definition at line 56 of file `ng_socketvar.h`.

6.273.2 Field Documentation

6.273.2.1 struct `ngpcb* ngsock::ctlsock`

Definition at line 59 of file `ng_socketvar.h`.

Referenced by `ng_detach_common()`, `ngs_rcvmsg()`, and `ngs_shutdown()`.

6.273.2.2 struct [ngpcb*](#) [ngsock::datasock](#)

Definition at line 58 of file [ng_socketvar.h](#).

Referenced by [ng_connect_data\(\)](#), [ng_detach_common\(\)](#), [ngs_connect\(\)](#), [ngs_disconnect\(\)](#), [ngs_rcvdata\(\)](#), and [ngs_shutdown\(\)](#).

6.273.2.3 int [ngsock::error](#)

Definition at line 63 of file [ng_socketvar.h](#).

Referenced by [ng_connect_data\(\)](#), [ng_socket_item_applied\(\)](#), and [ngc_send\(\)](#).

6.273.2.4 int [ngsock::flags](#)

Definition at line 60 of file [ng_socketvar.h](#).

Referenced by [ngs_disconnect\(\)](#), and [ngs_rcvmsg\(\)](#).

6.273.2.5 struct [mtx](#) [ngsock::mtx](#)

Definition at line 62 of file [ng_socketvar.h](#).

Referenced by [ng_connect_data\(\)](#), [ng_detach_common\(\)](#), [ng_socket_free_priv\(\)](#), [ng_socket_item_applied\(\)](#), [ngc_send\(\)](#), and [ngs_shutdown\(\)](#).

6.273.2.6 struct [ng_node*](#) [ngsock::node](#)

Definition at line 57 of file [ng_socketvar.h](#).

Referenced by [ng_bind\(\)](#), [ng_socket_free_priv\(\)](#), [ngc_send\(\)](#), [ngd_send\(\)](#), [ngs_connect\(\)](#), [ngs_disconnect\(\)](#), [ngs_rcvmsg\(\)](#), and [ngs_shutdown\(\)](#).

6.273.2.7 int [ngsock::refs](#)

Definition at line 61 of file [ng_socketvar.h](#).

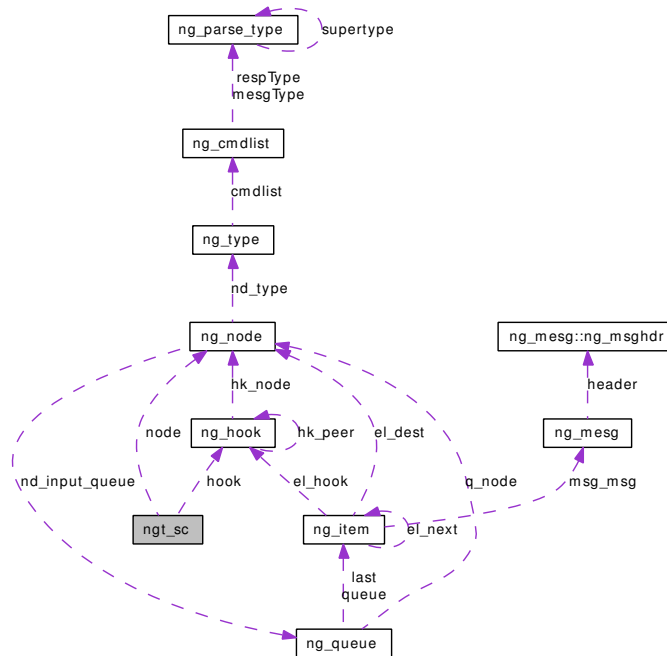
Referenced by [ng_attach_cntl\(\)](#), [ng_connect_data\(\)](#), and [ng_socket_free_priv\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_socketvar.h](#)

6.274 ngt_sc Struct Reference

Collaboration diagram for ngt_sc:



Data Fields

- tty * tp
- node_p node
- hook_p hook
- ifqueue outq
- mbuf * m
- short hotchar
- u_int flags
- callout chand

6.274.1 Detailed Description

Definition at line 87 of file ng_tty.c.

6.274.2 Field Documentation

6.274.2.1 struct callout `ngt_sc::chand`

Definition at line 95 of file ng_tty.c.

6.274.2.2 `u_int` `ngt_sc::flags`

Definition at line 94 of file `ng_tty.c`.

6.274.2.3 `hook_p` `ngt_sc::hook`

Definition at line 90 of file `ng_tty.c`.

6.274.2.4 `short` `ngt_sc::hotchar`

Definition at line 93 of file `ng_tty.c`.

6.274.2.5 `struct mbuf*` `ngt_sc::m`

Definition at line 92 of file `ng_tty.c`.

6.274.2.6 `node_p` `ngt_sc::node`

Definition at line 89 of file `ng_tty.c`.

6.274.2.7 `struct ifqueue` `ngt_sc::outq`

Definition at line 91 of file `ng_tty.c`.

6.274.2.8 `struct tty*` `ngt_sc::tp`

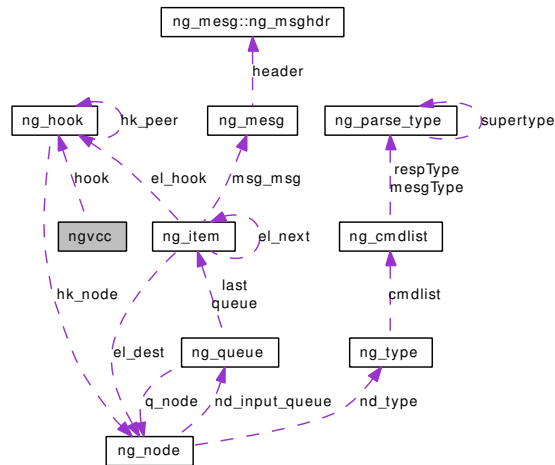
Definition at line 88 of file `ng_tty.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_tty.c`

6.275 ngvcc Struct Reference

Collaboration diagram for ngvcc:



Data Fields

- `uint16_t vpi`
- `uint16_t vci`
- `uint32_t flags`
- `hook_p hook`

6.275.1 Detailed Description

Definition at line 89 of file `ng_atm.c`.

6.275.2 Field Documentation

6.275.2.1 `uint32_t ngvcc::flags`

Definition at line 92 of file `ng_atm.c`.

Referenced by `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, and `ng_atm_disconnect()`.

6.275.2.2 `hook_p ngvcc::hook`

Definition at line 93 of file `ng_atm.c`.

Referenced by `ng_atm_connect()`, `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `ng_atm_disconnect()`, `ng_atm_event_func()`, `ng_atm_input()`, `ng_atm_newhook()`, `ng_atm_rcvdata()`, and `ng_atm_rcvmsg()`.

6.275.2.3 `uint16_t ngvcc::vci`

Definition at line 91 of file `ng_atm.c`.

Referenced by `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `ng_atm_disconnect()`, `ng_atm_event_func()`, `ng_atm_rcvdata()`, and `ng_atm_rcvmsg()`.

6.275.2.4 `uint16_t ngvcc::vpi`

Definition at line 90 of file `ng_atm.c`.

Referenced by `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `ng_atm_disconnect()`, `ng_atm_event_func()`, `ng_atm_rcvdata()`, and `ng_atm_rcvmsg()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_atm.c](#)

6.276 ngxxxstat Struct Reference

```
#include <ng_sample.h>
```

Data Fields

- [u_int32_t packets_in](#)
- [u_int32_t packets_out](#)

6.276.1 Detailed Description

Definition at line 71 of file `ng_sample.h`.

6.276.2 Field Documentation

6.276.2.1 [u_int32_t ngxxxstat::packets_in](#)

Definition at line 72 of file `ng_sample.h`.

6.276.2.2 [u_int32_t ngxxxstat::packets_out](#)

Definition at line 73 of file `ng_sample.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_sample.h](#)

6.277 nodeinfo Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [name](#) [NG_NODESIZ]
- char [type](#) [NG_TYPERIZ]
- [ng_ID_t id](#)
- [u_int32_t hooks](#)

6.277.1 Detailed Description

Definition at line 233 of file [ng_message.h](#).

6.277.2 Field Documentation

6.277.2.1 [u_int32_t nodeinfo::hooks](#)

Definition at line 237 of file [ng_message.h](#).

Referenced by [ng_generic_linkinfo_getLength\(\)](#), and [ng_generic_msg\(\)](#).

6.277.2.2 [ng_ID_t nodeinfo::id](#)

Definition at line 236 of file [ng_message.h](#).

Referenced by [ng_generic_msg\(\)](#).

6.277.2.3 [char nodeinfo::name](#)[NG_NODESIZ]

Definition at line 234 of file [ng_message.h](#).

Referenced by [ng_generic_msg\(\)](#).

6.277.2.4 [char nodeinfo::type](#)[NG_TYPERIZ]

Definition at line 235 of file [ng_message.h](#).

Referenced by [ng_generic_msg\(\)](#).

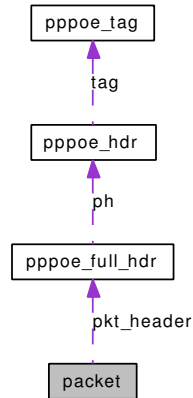
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.278 packet Union Reference

```
#include <ng_pppoe.h>
```

Collaboration diagram for packet:



Data Fields

- [pppoe_full_hdr pkt_header](#)
- [u_int8_t bytes](#) [2048]

6.278.1 Detailed Description

Definition at line 227 of file `ng_pppoe.h`.

6.278.2 Field Documentation

6.278.2.1 `u_int8_t packet::bytes`[2048]

Definition at line 229 of file `ng_pppoe.h`.

6.278.2.2 `struct pppoe_full_hdr packet::pkt_header`

Definition at line 228 of file `ng_pppoe.h`.

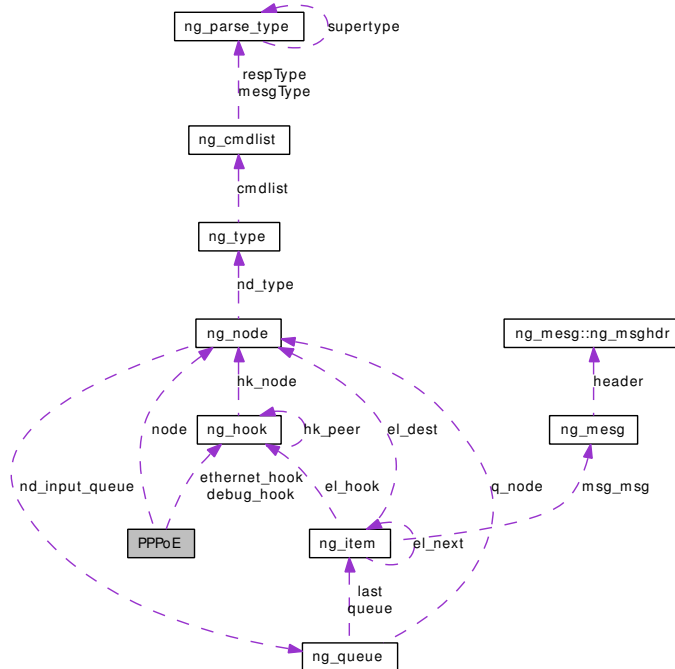
Referenced by `make_packet()`, `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, and `pppoe_start()`.

The documentation for this union was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.279 PPPoE Struct Reference

Collaboration diagram for PPPoE:



Data Fields

- [node_p](#) `node`
- [hook_p](#) `ethernet_hook`
- [hook_p](#) `debug_hook`
- `u_int` `packets_in`
- `u_int` `packets_out`
- `uint32_t` `flags`
- `ether_header` `eh`

6.279.1 Detailed Description

Definition at line 245 of file `ng_pppoe.c`.

6.279.2 Field Documentation

6.279.2.1 `hook_p` `PPPoE::debug_hook`

Definition at line 248 of file `ng_pppoe.c`.

6.279.2.2 `struct ether_header` `PPPoE::eh`

Definition at line 254 of file `ng_pppoe.c`.

6.279.2.3 [hook_p PPPoE::ethernet_hook](#)

Definition at line 247 of file `ng_pppoe.c`.

6.279.2.4 [uint32_t PPPoE::flags](#)

Definition at line 251 of file `ng_pppoe.c`.

6.279.2.5 [node_p PPPoE::node](#)

Definition at line 246 of file `ng_pppoe.c`.

6.279.2.6 [u_int PPPoE::packets_in](#)

Definition at line 249 of file `ng_pppoe.c`.

6.279.2.7 [u_int PPPoE::packets_out](#)

Definition at line 250 of file `ng_pppoe.c`.

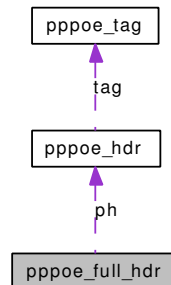
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.c](#)

6.280 pppoe_full_hdr Struct Reference

```
#include <ng_pppoe.h>
```

Collaboration diagram for pppoe_full_hdr:



Data Fields

- ether_header [eh](#)
- pppoe_hdr [ph](#)

6.280.1 Detailed Description

Definition at line 222 of file ng_pppoe.h.

6.280.2 Field Documentation

6.280.2.1 struct ether_header [pppoe_full_hdr::eh](#)

Definition at line 223 of file ng_pppoe.h.

Referenced by ng_pppoe_disconnect(), ng_pppoe_rcvdata(), pppoe_findsession(), and pppoe_start().

6.280.2.2 struct [pppoe_hdr](#) [pppoe_full_hdr::ph](#)

Definition at line 224 of file ng_pppoe.h.

Referenced by make_packet(), ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), pppoe_findsession(), and pppoe_start().

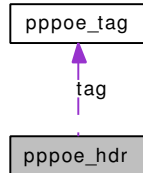
The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/ng_pppoe.h

6.281 pppoe_hdr Struct Reference

```
#include <ng_pppoe.h>
```

Collaboration diagram for pppoe_hdr:



Data Fields

- [u_int8_t ver](#):4
- [u_int8_t type](#):4
- [u_int8_t code](#)
- [u_int16_t sid](#)
- [u_int16_t length](#)
- [pppoe_tag tag](#) []

6.281.1 Detailed Description

Definition at line 212 of file ng_pppoe.h.

6.281.2 Field Documentation

6.281.2.1 u_int8_t pppoe_hdr::code

Definition at line 215 of file ng_pppoe.h.

Referenced by [ng_pppoe_disconnect\(\)](#), [ng_pppoe_rcvdata\(\)](#), [ng_pppoe_rcvmsg\(\)](#), and [pppoe_start\(\)](#).

6.281.2.2 u_int16_t pppoe_hdr::length

Definition at line 217 of file ng_pppoe.h.

Referenced by [make_packet\(\)](#), [next_tag\(\)](#), and [ng_pppoe_rcvdata\(\)](#).

6.281.2.3 u_int16_t pppoe_hdr::sid

Definition at line 216 of file ng_pppoe.h.

Referenced by [ng_pppoe_rcvdata\(\)](#), and [pppoe_findsession\(\)](#).

6.281.2.4 struct pppoe_tag pppoe_hdr::tag[]

Definition at line 218 of file ng_pppoe.h.

Referenced by [get_tag\(\)](#), [make_packet\(\)](#), [next_tag\(\)](#), and [scan_tags\(\)](#).

6.281.2.5 `u_int8_t pppoe_hdr::type`

Definition at line 214 of file `ng_pppoe.h`.

6.281.2.6 `u_int8_t pppoe_hdr::ver`

Definition at line 213 of file `ng_pppoe.h`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.282 pppoe_tag Struct Reference

```
#include <ng_pppoe.h>
```

Data Fields

- [u_int16_t tag_type](#)
- [u_int16_t tag_len](#)
- [char tag_data \[\]](#)

6.282.1 Detailed Description

Definition at line 206 of file `ng_pppoe.h`.

6.282.2 Field Documentation

6.282.2.1 [char pppoe_tag::tag_data\[\]](#)

Definition at line 209 of file `ng_pppoe.h`.

Referenced by `ng_pppoe_disconnect()`, `pppoe_finduniq()`, `pppoe_match_svc()`, and `send_acname()`.

6.282.2.2 [u_int16_t pppoe_tag::tag_len](#)

Definition at line 208 of file `ng_pppoe.h`.

Referenced by `get_tag()`, `ng_pppoe_disconnect()`, `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, `pppoe_match_svc()`, `scan_tags()`, and `send_acname()`.

6.282.2.3 [u_int16_t pppoe_tag::tag_type](#)

Definition at line 207 of file `ng_pppoe.h`.

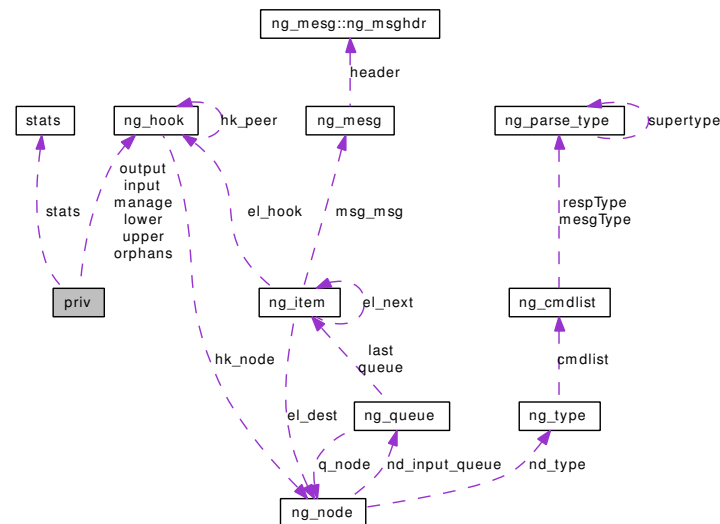
Referenced by `get_tag()`, `ng_pppoe_disconnect()`, `ng_pppoe_rcvmsg()`, and `scan_tags()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.h](#)

6.283 priv Struct Reference

Collaboration diagram for priv:



Data Fields

- `ifnet * ifp`
- `hook_p input`
- `hook_p orphans`
- `hook_p output`
- `hook_p manage`
- `uint64_t in_packets`
- `uint64_t in_errors`
- `uint64_t out_packets`
- `uint64_t out_errors`
- `hook_p upper`
- `hook_p lower`
- `sscfu * sscf`
- `int enabled`
- `sscop * sscop`
- `int flow`
- `stats stats`
- `uni * uni`

6.283.1 Detailed Description

Definition at line 102 of file `ng_atm.c`.

6.283.2 Field Documentation

6.283.2.1 `int priv::enabled`

Definition at line 68 of file `ng_sscfu.c`.

Referenced by `ng_sscfu_disconnect()`, `ng_sscfu_rcvlower()`, `ng_sscfu_rcvmsg()`, `ng_sscfu_rcvupper()`, `ng_sscop_disconnect()`, `ng_sscop_rcvlower()`, `ng_sscop_rcvmanage()`, `ng_sscop_rcvmsg()`, `ng_sscop_rcvupper()`, `ng_uni_rcvlower()`, `ng_uni_rcvmsg()`, `ng_uni_rcvupper()`, and `text_status()`.

6.283.2.2 `int priv::flow`

Definition at line 97 of file `ng_sscop.c`.

Referenced by `flow_upper()`, `ng_sscop_rcvmsg()`, and `sscop_send_upper()`.

6.283.2.3 `struct ifnet* priv::ifp`

Definition at line 103 of file `ng_atm.c`.

Referenced by `cpcs_term()`, `ng_atm_cpcs_init()`, `ng_atm_detach()`, `ng_atm_disconnect()`, `ng_atm_rcvdata()`, `ng_atm_rcvmsg()`, `ng_atm_shutdown()`, and `text_status()`.

6.283.2.4 `uint64_t priv::in_errors`

Definition at line 109 of file `ng_atm.c`.

Referenced by `ng_atm_input()`, `ng_atm_input_orphans()`, and `ng_atm_rcvmsg()`.

6.283.2.5 `uint64_t priv::in_packets`

Definition at line 108 of file `ng_atm.c`.

Referenced by `ng_atm_input()`, `ng_atm_input_orphans()`, and `ng_atm_rcvmsg()`.

6.283.2.6 `hook_p priv::input`

Definition at line 104 of file `ng_atm.c`.

Referenced by `ng_atm_disconnect()`, `ng_atm_input()`, and `ng_atm_newhook()`.

6.283.2.7 `hook_p priv::lower`

Definition at line 66 of file `ng_sscfu.c`.

Referenced by `ng_sscfu_disconnect()`, `ng_sscfu_newhook()`, `ng_sscfu_rcvupper()`, `ng_sscop_disconnect()`, `ng_sscop_newhook()`, `ng_sscop_rcvmsg()`, `ng_uni_disconnect()`, `ng_uni_newhook()`, `sscfu_send_lower()`, `sscop_send_lower()`, `text_status()`, and `uni_saal_output()`.

6.283.2.8 `hook_p priv::manage`

Definition at line 107 of file `ng_atm.c`.

Referenced by `ng_atm_event_func()`, `ng_atm_newhook()`, `ng_sscop_disconnect()`, `ng_sscop_newhook()`, `sscop_send_manage()`, and `text_status()`.

6.283.2.9 `hook_p priv::orphans`

Definition at line 105 of file `ng_atm.c`.

Referenced by `ng_atm_disconnect()`, `ng_atm_input_orphans()`, and `ng_atm_newhook()`.

6.283.2.10 `uint64_t priv::out_errors`

Definition at line 111 of file `ng_atm.c`.

Referenced by `ng_atm_rcvdata()`, and `ng_atm_rcvmsg()`.

6.283.2.11 `uint64_t priv::out_packets`

Definition at line 110 of file `ng_atm.c`.

Referenced by `ng_atm_rcvdata()`, and `ng_atm_rcvmsg()`.

6.283.2.12 `hook_p priv::output`

Definition at line 106 of file `ng_atm.c`.

Referenced by `ng_atm_disconnect()`, `ng_atm_newhook()`, and `ng_atm_output()`.

6.283.2.13 `struct sscfu* priv::sscf`

Definition at line 67 of file `ng_sscfu.c`.

Referenced by `ng_sscfu_disconnect()`, `ng_sscfu_rcvlower()`, `ng_sscfu_rcvmsg()`, `ng_sscfu_rcvupper()`, `ng_sscfu_shutdown()`, and `text_status()`.

6.283.2.14 `struct sscop* priv::sscop`

Definition at line 95 of file `ng_sscop.c`.

Referenced by `flow_lower()`, `flow_upper()`, `ng_sscop_disconnect()`, `ng_sscop_rcvlower()`, `ng_sscop_rcvmanage()`, `ng_sscop_rcvmsg()`, `ng_sscop_rcvupper()`, `ng_sscop_shutdown()`, `sscop_send_upper()`, and `text_status()`.

6.283.2.15 `struct stats priv::stats`

Definition at line 98 of file `ng_sscop.c`.

Referenced by `ng_sscop_rcvlower()`, `ng_sscop_rcvmsg()`, `sscop_send_lower()`, `sscop_send_manage()`, `sscop_send_upper()`, and `text_status()`.

6.283.2.16 `struct uni* priv::uni`

Definition at line 74 of file `ng_uni.c`.

Referenced by `dump_saal_signal()`, `ng_uni_rcvlower()`, `ng_uni_rcvmsg()`, `ng_uni_rcvupper()`, `ng_uni_shutdown()`, `text_status()`, and `uni_saal_output()`.

6.283.2.17 `hook_p priv::upper`

Definition at line 65 of file `ng_sscfu.c`.

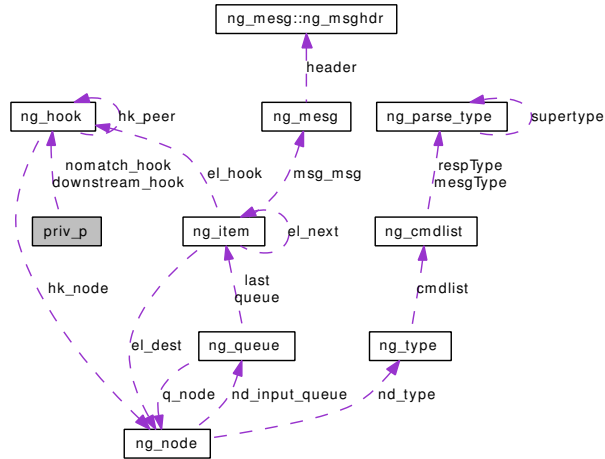
Referenced by `ng_sscfu_disconnect()`, `ng_sscfu_newhook()`, `ng_sscfu_rcvlower()`, `ng_sscop_disconnect()`, `ng_sscop_newhook()`, `ng_sscop_rcvlower()`, `ng_sscop_rcvmsg()`, `ng_uni_disconnect()`, `ng_uni_newhook()`, `sscfu_send_upper()`, `sscop_send_upper()`, `text_status()`, and `uni_uni_output()`.

The documentation for this struct was generated from the following files:

- `/usr/src/sys/netgraph/atm/ng_atm.c`
- `/usr/src/sys/netgraph/atm/sscfu/ng_sscfu.c`
- `/usr/src/sys/netgraph/atm/sscop/ng_sscop.c`
- `/usr/src/sys/netgraph/atm/uni/ng_uni.c`

6.284 priv_p Struct Reference

Collaboration diagram for priv_p:



Data Fields

- `hook_p` `downstream_hook`
- `hook_p` `nomatch_hook`
- filterhead `hashtable` [HASHSIZE]
- `u_int32_t` `nent`

6.284.1 Detailed Description

Definition at line 139 of file `ng_vlan.c`.

6.284.2 Field Documentation

6.284.2.1 `hook_p` `priv_p::downstream_hook`

Definition at line 140 of file `ng_vlan.c`.

Referenced by `ng_vlan_disconnect()`, `ng_vlan_newhook()`, `ng_vlan_rcvdata()`, and `ng_vlan_rcvmsg()`.

6.284.2.2 `struct filterhead` `priv_p::hashtable`[HASHSIZE]

Definition at line 142 of file `ng_vlan.c`.

Referenced by `ng_vlan_findextry()`, and `ng_vlan_rcvmsg()`.

6.284.2.3 `u_int32_t` `priv_p::nent`

Definition at line 143 of file `ng_vlan.c`.

Referenced by `ng_vlan_disconnect()`, and `ng_vlan_rcvmsg()`.

6.284.2.4 [hook_p_priv_p::nomatch_hook](#)

Definition at line 141 of file `ng_vlan.c`.

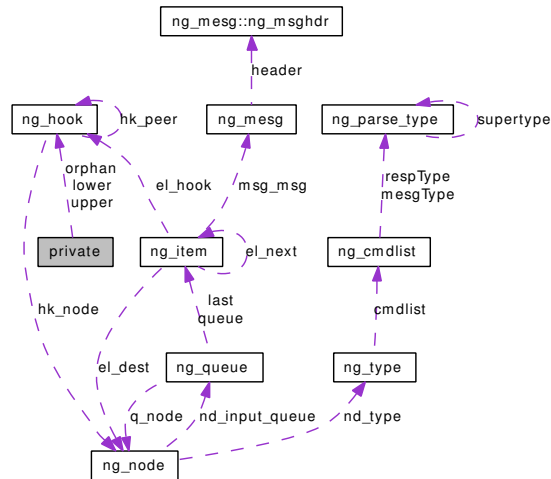
Referenced by `ng_vlan_disconnect()`, `ng_vlan_newhook()`, `ng_vlan_rcvdata()`, and `ng_vlan_rcvmsg()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_vlan.c](#)

6.285 private Struct Reference

Collaboration diagram for private:



Data Fields

- ifnet * ifp
- hook_p upper
- hook_p lower
- hook_p orphan
- u_char autoSrcAddr
- u_char promisc
- u_long hwassist
- u_int flags
- ifnet * ifp
- u_char lowerOrphan

6.285.1 Detailed Description

Definition at line 74 of file ng_ether.c.

6.285.2 Field Documentation

6.285.2.1 u_char private::autoSrcAddr

Definition at line 79 of file ng_ether.c.

6.285.2.2 u_int private::flags

Definition at line 82 of file ng_ether.c.

6.285.2.3 `u_long private::hwassist`

Definition at line 81 of file `ng_ether.c`.

6.285.2.4 `struct ifnet* private::ifp`

Definition at line 97 of file `ng_gif.c`.

6.285.2.5 `struct ifnet* private::ifp`

Definition at line 75 of file `ng_ether.c`.

6.285.2.6 `hook_p private::lower`

Definition at line 77 of file `ng_ether.c`.

6.285.2.7 `u_char private::lowerOrphan`

Definition at line 99 of file `ng_gif.c`.

6.285.2.8 `hook_p private::orphan`

Definition at line 78 of file `ng_ether.c`.

6.285.2.9 `u_char private::promisc`

Definition at line 80 of file `ng_ether.c`.

6.285.2.10 `hook_p private::upper`

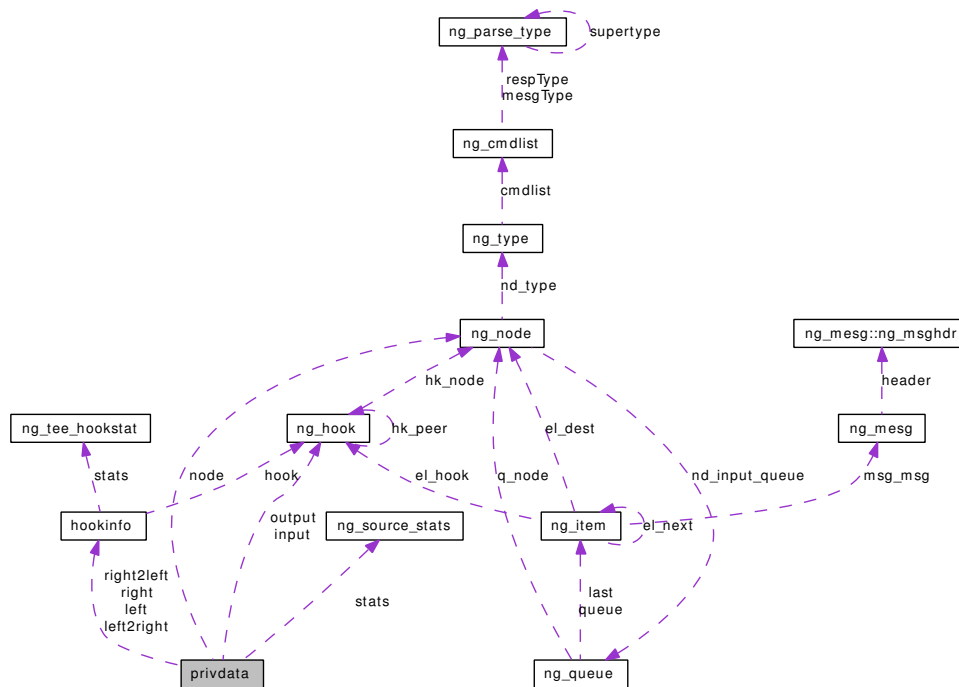
Definition at line 76 of file `ng_ether.c`.

The documentation for this struct was generated from the following files:

- `/usr/src/sys/netgraph/ng_ether.c`
- `/usr/src/sys/netgraph/ng_gif.c`

6.286 privdata Struct Reference

Collaboration diagram for privdata:



Data Fields

- `node_p` node
- `hook_p` input
- `hook_p` output
- `ng_source_stats` stats
- ifqueue `snd_queue`
- ifnet * `output_ifp`
- callout `intr_ch`
- `uint64_t` packets
- `uint32_t` queueOctets
- `hookinfo` left
- `hookinfo` right
- `hookinfo` left2right
- `hookinfo` right2left

6.286.1 Detailed Description

Definition at line 81 of file `ng_source.c`.

6.286.2 Field Documentation

6.286.2.1 `hook_p privdata::input`

Definition at line 83 of file `ng_source.c`.

6.286.2.2 `struct callout privdata::intr_ch`

Definition at line 88 of file `ng_source.c`.

6.286.2.3 `struct hookinfo privdata::left`

Definition at line 74 of file `ng_tee.c`.

6.286.2.4 `struct hookinfo privdata::left2right`

Definition at line 76 of file `ng_tee.c`.

6.286.2.5 `node_p privdata::node`

Definition at line 82 of file `ng_source.c`.

6.286.2.6 `hook_p privdata::output`

Definition at line 84 of file `ng_source.c`.

6.286.2.7 `struct ifnet* privdata::output_ifp`

Definition at line 87 of file `ng_source.c`.

6.286.2.8 `uint64_t privdata::packets`

Definition at line 89 of file `ng_source.c`.

6.286.2.9 `uint32_t privdata::queueOctets`

Definition at line 90 of file `ng_source.c`.

6.286.2.10 `struct hookinfo privdata::right`

Definition at line 75 of file `ng_tee.c`.

6.286.2.11 `struct hookinfo privdata::right2left`

Definition at line 77 of file `ng_tee.c`.

6.286.2.12 struct ifqueue [privdata::snd_queue](#)

Definition at line 86 of file [ng_source.c](#).

6.286.2.13 struct [ng_source_stats](#) [privdata::stats](#)

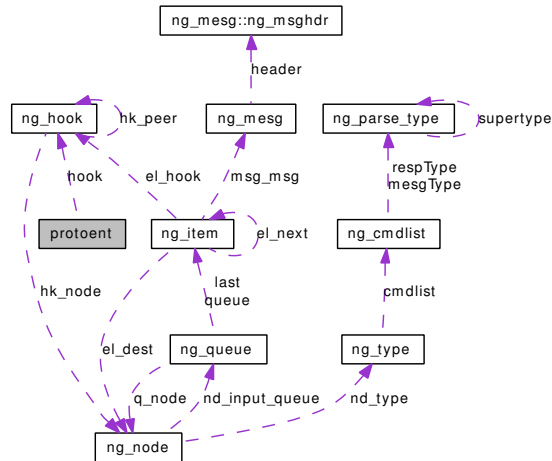
Definition at line 85 of file [ng_source.c](#).

The documentation for this struct was generated from the following files:

- [/usr/src/sys/netgraph/ng_source.c](#)
- [/usr/src/sys/netgraph/ng_tee.c](#)

6.287 protoent Struct Reference

Collaboration diagram for protoent:



Data Fields

- [hook_p](#) hook
- [u_short](#) af

6.287.1 Detailed Description

Definition at line 96 of file ng_cisco.c.

6.287.2 Field Documentation

6.287.2.1 [u_short](#) protoent::af

Definition at line 98 of file ng_cisco.c.

Referenced by [cisco_disconnect\(\)](#), and [cisco_rcvdata\(\)](#).

6.287.2.2 [hook_p](#) protoent::hook

Definition at line 97 of file ng_cisco.c.

Referenced by [cisco_disconnect\(\)](#), and [cisco_input\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_cisco.c](#)

6.288 rfcomm_cmd_hdr Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t address](#)
- [u_int8_t control](#)
- [u_int8_t length](#)
- [u_int8_t fcs](#)

6.288.1 Detailed Description

Definition at line 128 of file `ng_btsocket_rfcomm.h`.

6.288.2 Field Documentation

6.288.2.1 [u_int8_t rfcomm_cmd_hdr::address](#)

Definition at line 130 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_send_command()`.

6.288.2.2 [u_int8_t rfcomm_cmd_hdr::control](#)

Definition at line 131 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_send_command()`.

6.288.2.3 [u_int8_t rfcomm_cmd_hdr::fcs](#)

Definition at line 133 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_send_command()`.

6.288.2.4 [u_int8_t rfcomm_cmd_hdr::length](#)

Definition at line 132 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_send_command()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h`

6.289 rfcomm_frame_hdr Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t address](#)
- [u_int8_t control](#)
- [u_int8_t length](#)

6.289.1 Detailed Description

Definition at line 120 of file `ng_btsocket_rfcomm.h`.

6.289.2 Field Documentation

6.289.2.1 [u_int8_t rfcomm_frame_hdr::address](#)

Definition at line 122 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_frame()`, and `ng_btsocket_rfcomm_send_uih()`.

6.289.2.2 [u_int8_t rfcomm_frame_hdr::control](#)

Definition at line 123 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_frame()`, and `ng_btsocket_rfcomm_send_uih()`.

6.289.2.3 [u_int8_t rfcomm_frame_hdr::length](#)

Definition at line 124 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_frame()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.290 rfcomm_mcc_hdr Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t type](#)
- [u_int8_t length](#)

6.290.1 Detailed Description

Definition at line 137 of file `ng_btsocket_rfcomm.h`.

6.290.2 Field Documentation

6.290.2.1 [u_int8_t rfcomm_mcc_hdr::length](#)

Definition at line 140 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_mcc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_send_msc()`, and `ng_btsocket_rfcomm_send_pn()`.

6.290.2.2 [u_int8_t rfcomm_mcc_hdr::type](#)

Definition at line 139 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_mcc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_send_msc()`, and `ng_btsocket_rfcomm_send_pn()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.291 rfcomm_mcc_msc Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t address](#)
- [u_int8_t modem](#)

6.291.1 Detailed Description

Definition at line 144 of file `ng_btsocket_rfcomm.h`.

6.291.2 Field Documentation

6.291.2.1 [u_int8_t rfcomm_mcc_msc::address](#)

Definition at line 146 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_msc()`.

6.291.2.2 [u_int8_t rfcomm_mcc_msc::modem](#)

Definition at line 147 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_msc()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.292 rfcomm_mcc_pn Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t dlc](#)
- [u_int8_t flow_control](#)
- [u_int8_t priority](#)
- [u_int8_t ack_timer](#)
- [u_int16_t mtu](#)
- [u_int8_t max_retrans](#)
- [u_int8_t credits](#)

6.292.1 Detailed Description

Definition at line 170 of file ng_btsocket_rfcomm.h.

6.292.2 Field Documentation

6.292.2.1 [u_int8_t rfcomm_mcc_pn::ack_timer](#)

Definition at line 175 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_receive_pn\(\)](#).

6.292.2.2 [u_int8_t rfcomm_mcc_pn::credits](#)

Definition at line 178 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_receive_pn\(\)](#), and [ng_btsocket_rfcomm_send_credits\(\)](#).

6.292.2.3 [u_int8_t rfcomm_mcc_pn::dlci](#)

Definition at line 172 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_receive_pn\(\)](#).

6.292.2.4 [u_int8_t rfcomm_mcc_pn::flow_control](#)

Definition at line 173 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_receive_pn\(\)](#).

6.292.2.5 [u_int8_t rfcomm_mcc_pn::max_retrans](#)

Definition at line 177 of file ng_btsocket_rfcomm.h.

Referenced by [ng_btsocket_rfcomm_receive_pn\(\)](#).

6.292.2.6 `u_int16_t rfcomm_mcc_pn::mtu`

Definition at line 176 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_pn()`.

6.292.2.7 `u_int8_t rfcomm_mcc_pn::priority`

Definition at line 174 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_pn()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h`

6.293 rfcomm_mcc_rls Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t address](#)
- [u_int8_t status](#)

6.293.1 Detailed Description

Definition at line 163 of file `ng_btsocket_rfcomm.h`.

6.293.2 Field Documentation

6.293.2.1 [u_int8_t rfcomm_mcc_rls::address](#)

Definition at line 165 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rls()`.

6.293.2.2 [u_int8_t rfcomm_mcc_rls::status](#)

Definition at line 166 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rls()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.294 rfcomm_mcc_rpn Struct Reference

```
#include <ng_btsocket_rfcomm.h>
```

Data Fields

- [u_int8_t dlc](#)
- [u_int8_t bit_rate](#)
- [u_int8_t line_settings](#)
- [u_int8_t flow_control](#)
- [u_int8_t xon_char](#)
- [u_int8_t xoff_char](#)
- [u_int16_t param_mask](#)

6.294.1 Detailed Description

Definition at line 151 of file `ng_btsocket_rfcomm.h`.

6.294.2 Field Documentation

6.294.2.1 [u_int8_t rfcomm_mcc_rpn::bit_rate](#)

Definition at line 154 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.2 [u_int8_t rfcomm_mcc_rpn::dlci](#)

Definition at line 153 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.3 [u_int8_t rfcomm_mcc_rpn::flow_control](#)

Definition at line 156 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.4 [u_int8_t rfcomm_mcc_rpn::line_settings](#)

Definition at line 155 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.5 [u_int16_t rfcomm_mcc_rpn::param_mask](#)

Definition at line 159 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.6 `u_int8_t rfcomm_mcc_rpn::xoff_char`

Definition at line 158 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

6.294.2.7 `u_int8_t rfcomm_mcc_rpn::xon_char`

Definition at line 157 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_receive_rpn()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h](#)

6.295 sa_tag Struct Reference

```
#include <ng_ksocket.h>
```

Data Fields

- [m_tag tag](#)
- [ng_ID_t id](#)
- [sockaddr sa](#)

6.295.1 Detailed Description

Definition at line 101 of file `ng_ksocket.h`.

6.295.2 Field Documentation

6.295.2.1 `ng_ID_t sa_tag::id`

Definition at line 103 of file `ng_ksocket.h`.

Referenced by `ng_ksocket_incoming2()`, and `ng_ksocket_rcvdata()`.

6.295.2.2 `struct sockaddr sa_tag::sa`

Definition at line 104 of file `ng_ksocket.h`.

Referenced by `ng_ksocket_incoming2()`, and `ng_ksocket_rcvdata()`.

6.295.2.3 `struct m_tag sa_tag::tag`

Definition at line 102 of file `ng_ksocket.h`.

Referenced by `ng_ksocket_incoming2()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_ksocket.h](#)

6.296 segment Struct Reference

Data Fields

- `u_char` [mask](#)
- `u_char` [shift](#)
- `u_char` [width](#)

6.296.1 Detailed Description

Definition at line 103 of file `ng_frame_relay.c`.

6.296.2 Field Documentation

6.296.2.1 `u_char` [segment::mask](#)

Definition at line 104 of file `ng_frame_relay.c`.

6.296.2.2 `u_char` [segment::shift](#)

Definition at line 105 of file `ng_frame_relay.c`.

6.296.2.3 `u_char` [segment::width](#)

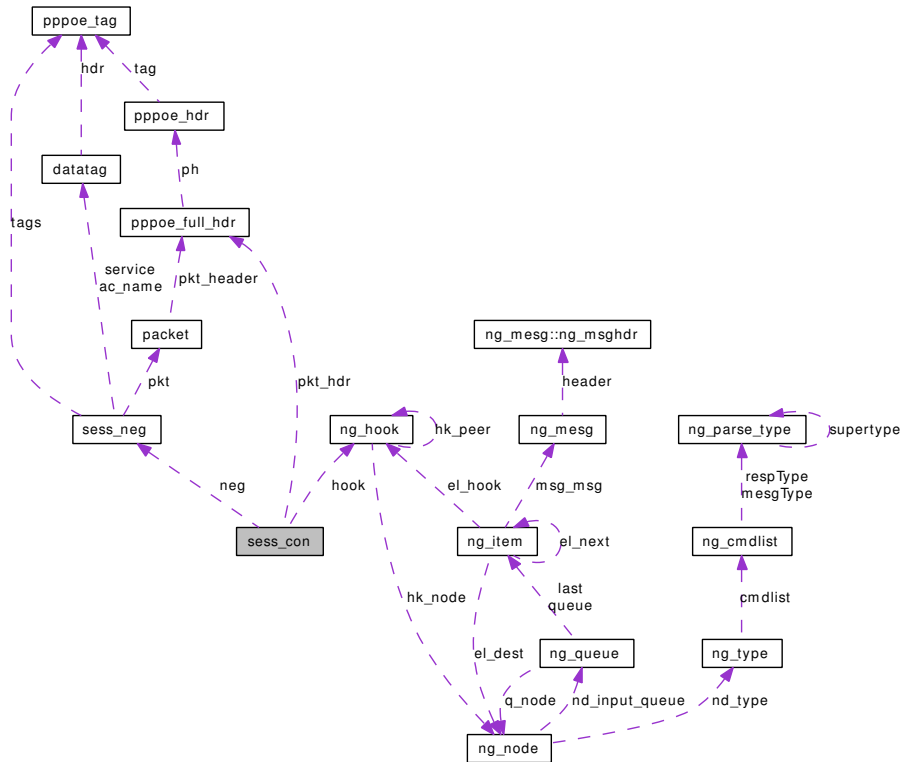
Definition at line 106 of file `ng_frame_relay.c`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_frame_relay.c`

6.297 sess_con Struct Reference

Collaboration diagram for sess_con:



Data Fields

- [hook_p](#) hook
- [uint16_t](#) Session_ID
- [enum](#) state state
- [ng_ID_t](#) creator
- [pppoe_full_hdr](#) pkt_hdr
- [negp](#) neg

6.297.1 Detailed Description

Definition at line 230 of file ng_pppoe.c.

6.297.2 Field Documentation

6.297.2.1 [ng_ID_t](#) sess_con::creator

Definition at line 234 of file ng_pppoe.c.

Referenced by [ng_pppoe_rcvmsg\(\)](#), [pppoe_send_event\(\)](#), [send_acname\(\)](#), and [send_sessionid\(\)](#).

6.297.2.2 [hook_p sess_con::hook](#)

Definition at line 231 of file ng_pppoe.c.

Referenced by ng_pppoe_sendpacket(), pppoe_send_event(), send_acname(), and send_sessionid().

6.297.2.3 [negp sess_con::neg](#)

Definition at line 236 of file ng_pppoe.c.

Referenced by init_tags(), insert_tag(), make_packet(), ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), ng_pppoe_sendpacket(), pppoe_find_svc(), pppoe_match_svc(), pppoe_start(), and pppoe_ticker().

6.297.2.4 [struct pppoe_full_hdr sess_con::pkt_hdr](#)

Definition at line 235 of file ng_pppoe.c.

Referenced by ng_pppoe_disconnect(), ng_pppoe_rcvdata(), and pppoe_findsession().

6.297.2.5 [uint16_t sess_con::Session_ID](#)

Definition at line 232 of file ng_pppoe.c.

Referenced by get_new_sid(), make_packet(), ng_pppoe_sendpacket(), pppoe_findsession(), pppoe_send_event(), pppoe_start(), pppoe_ticker(), scan_tags(), send_acname(), and send_sessionid().

6.297.2.6 [enum state sess_con::state](#)

Definition at line 233 of file ng_pppoe.c.

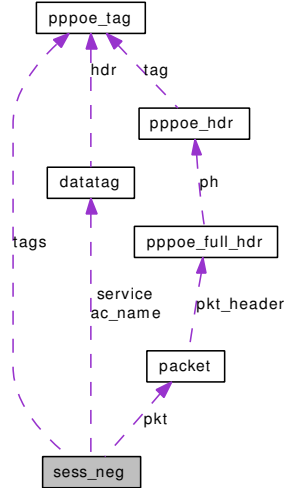
Referenced by ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), ng_pppoe_sendpacket(), pppoe_broadcast_padi(), pppoe_find_svc(), pppoe_findsession(), pppoe_match_svc(), pppoe_start(), and pppoe_ticker().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.c](#)

6.298 sess_neg Struct Reference

Collaboration diagram for sess_neg:



Data Fields

- `mbuf * m`
- `packet * pkt`
- callout `handle`
- `u_int timeout`
- `u_int numtags`
- `pppoe_tag * tags` [NUMTAGS]
- `u_int service_len`
- `u_int ac_name_len`
- `datatag service`
- `datatag ac_name`

6.298.1 Detailed Description

Definition at line 212 of file `ng_pppoe.c`.

6.298.2 Field Documentation

6.298.2.1 struct `datatag sess_neg::ac_name`

Definition at line 223 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_rcvdata()`, and `ng_pppoe_rcvmsg()`.

6.298.2.2 `u_int sess_neg::ac_name_len`

Definition at line 220 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_rcvmsg()`.

6.298.2.3 struct callout sess_neg::handle

Definition at line 215 of file ng_pppoe.c.

Referenced by ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_sendpacket(), and pppoe_ticker().

6.298.2.4 struct mbuf* sess_neg::m

Definition at line 213 of file ng_pppoe.c.

Referenced by make_packet(), ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_sendpacket(), and pppoe_ticker().

6.298.2.5 u_int sess_neg::numtags

Definition at line 217 of file ng_pppoe.c.

Referenced by init_tags(), insert_tag(), and make_packet().

6.298.2.6 union packet* sess_neg::pkt

Definition at line 214 of file ng_pppoe.c.

Referenced by make_packet(), ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), and pppoe_start().

6.298.2.7 struct datatag sess_neg::service

Definition at line 222 of file ng_pppoe.c.

Referenced by ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), pppoe_find_svc(), pppoe_match_svc(), and pppoe_start().

6.298.2.8 u_int sess_neg::service_len

Definition at line 219 of file ng_pppoe.c.

Referenced by ng_pppoe_rcvmsg(), pppoe_find_svc(), and pppoe_match_svc().

6.298.2.9 struct pppoe_tag* sess_neg::tags[NUMTAGS]

Definition at line 218 of file ng_pppoe.c.

Referenced by insert_tag(), and make_packet().

6.298.2.10 u_int sess_neg::timeout

Definition at line 216 of file ng_pppoe.c.

Referenced by ng_pppoe_rcvdata(), ng_pppoe_sendpacket(), and pppoe_ticker().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netgraph/ng_pppoe.c

6.299 sockaddr_hci Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- u_char [hci_len](#)
- u_char [hci_family](#)
- char [hci_node](#) [32]

6.299.1 Detailed Description

Definition at line 49 of file `ng_btsocket.h`.

6.299.2 Field Documentation

6.299.2.1 u_char [sockaddr_hci::hci_family](#)

Definition at line 51 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_bind()`, `ng_btsocket_hci_raw_connect()`, and `ng_btsocket_hci_raw_node_rcvdata()`.

6.299.2.2 u_char [sockaddr_hci::hci_len](#)

Definition at line 50 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_bind()`, `ng_btsocket_hci_raw_connect()`, and `ng_btsocket_hci_raw_node_rcvdata()`.

6.299.2.3 char [sockaddr_hci::hci_node](#)[32]

Definition at line 52 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_data_input()`, `ng_btsocket_hci_raw_node_rcvdata()`, `ng_btsocket_hci_raw_output()`, `ng_btsocket_hci_raw_send()`, and `ng_btsocket_hci_raw_sockaddr()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.300 sockaddr_l2cap Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- [u_char l2cap_len](#)
- [u_char l2cap_family](#)
- [u_int16_t l2cap_psm](#)
- [bdaddr_t l2cap_bdaddr](#)

6.300.1 Detailed Description

Definition at line 206 of file ng_btsocket.h.

6.300.2 Field Documentation

6.300.2.1 [bdaddr_t sockaddr_l2cap::l2cap_bdaddr](#)

Definition at line 210 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_sockaddr\(\)](#), [ng_btsocket_l2cap_sockaddr\(\)](#), and [ng_btsocket_rfcomm_session_create\(\)](#).

6.300.2.2 [u_char sockaddr_l2cap::l2cap_family](#)

Definition at line 208 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_bind\(\)](#), [ng_btsocket_l2cap_connect\(\)](#), [ng_btsocket_l2cap_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_bind\(\)](#), [ng_btsocket_l2cap_raw_connect\(\)](#), [ng_btsocket_l2cap_raw_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_sockaddr\(\)](#), [ng_btsocket_l2cap_sockaddr\(\)](#), and [ng_btsocket_rfcomm_session_create\(\)](#).

6.300.2.3 [u_char sockaddr_l2cap::l2cap_len](#)

Definition at line 207 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_bind\(\)](#), [ng_btsocket_l2cap_connect\(\)](#), [ng_btsocket_l2cap_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_bind\(\)](#), [ng_btsocket_l2cap_raw_connect\(\)](#), [ng_btsocket_l2cap_raw_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_sockaddr\(\)](#), [ng_btsocket_l2cap_sockaddr\(\)](#), and [ng_btsocket_rfcomm_session_create\(\)](#).

6.300.2.4 [u_int16_t sockaddr_l2cap::l2cap_psm](#)

Definition at line 209 of file ng_btsocket.h.

Referenced by [ng_btsocket_l2cap_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_peeraddr\(\)](#), [ng_btsocket_l2cap_raw_sockaddr\(\)](#), [ng_btsocket_l2cap_sockaddr\(\)](#), and [ng_btsocket_rfcomm_session_create\(\)](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h](#)

6.301 sockaddr_ng Struct Reference

```
#include <ng_socket.h>
```

Data Fields

- unsigned char [sg_len](#)
- sa_family_t [sg_family](#)
- char [sg_data](#) [14]

6.301.1 Detailed Description

Definition at line 62 of file `ng_socket.h`.

6.301.2 Field Documentation

6.301.2.1 char [sockaddr_ng::sg_data](#)[14]

Definition at line 65 of file `ng_socket.h`.

Referenced by `ng_bind()`, `ng_connect_data()`, `ng_setsockaddr()`, `ngc_send()`, `ngd_send()`, and `ngs_rcvdata()`.

6.301.2.2 sa_family_t [sockaddr_ng::sg_family](#)

Definition at line 64 of file `ng_socket.h`.

Referenced by `ng_setsockaddr()`, and `ngs_rcvdata()`.

6.301.2.3 unsigned char [sockaddr_ng::sg_len](#)

Definition at line 63 of file `ng_socket.h`.

Referenced by `ng_bind()`, `ng_setsockaddr()`, `ngc_send()`, `ngd_send()`, and `ngs_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_socket.h](#)

6.302 sockaddr_rfcomm Struct Reference

```
#include <ng_btsocket.h>
```

Data Fields

- u_char [rfcomm_len](#)
- u_char [rfcomm_family](#)
- bdaddr_t [rfcomm_bdaddr](#)
- u_int8_t [rfcomm_channel](#)

6.302.1 Detailed Description

Definition at line 301 of file `ng_btsocket.h`.

6.302.2 Field Documentation

6.302.2.1 bdaddr_t [sockaddr_rfcomm::rfcomm_bdaddr](#)

Definition at line 304 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_peeraddr()`, and `ng_btsocket_rfcomm_sockaddr()`.

6.302.2.2 u_int8_t [sockaddr_rfcomm::rfcomm_channel](#)

Definition at line 305 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_peeraddr()`, and `ng_btsocket_rfcomm_sockaddr()`.

6.302.2.3 u_char [sockaddr_rfcomm::rfcomm_family](#)

Definition at line 303 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_bind()`, `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_peeraddr()`, and `ng_btsocket_rfcomm_sockaddr()`.

6.302.2.4 u_char [sockaddr_rfcomm::rfcomm_len](#)

Definition at line 302 of file `ng_btsocket.h`.

Referenced by `ng_btsocket_rfcomm_bind()`, `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_peeraddr()`, and `ng_btsocket_rfcomm_sockaddr()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h`

6.303 sscfu_arg Struct Reference

```
#include <ng_sscfu.h>
```

Data Fields

- `uint32_t sig`
- `u_char data []`

6.303.1 Detailed Description

Definition at line 64 of file `ng_sscfu.h`.

6.303.2 Field Documentation

6.303.2.1 `u_char sscfu_arg::data[]`

Definition at line 66 of file `ng_sscfu.h`.

6.303.2.2 `uint32_t sscfu_arg::sig`

Definition at line 65 of file `ng_sscfu.h`.

Referenced by `sscfu_send_upper()`, and `uni_saal_output()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_sscfu.h`

6.304 sscop_arg Struct Reference

```
#include <ng_sscop.h>
```

Data Fields

- uint32_t [sig](#)
- uint32_t [arg](#)
- u_char [data](#) []

6.304.1 Detailed Description

Definition at line 94 of file `ng_sscop.h`.

6.304.2 Field Documentation

6.304.2.1 uint32_t [sscop_arg::arg](#)

Definition at line 96 of file `ng_sscop.h`.

Referenced by `sscfu_send_lower()`, and `sscop_send_upper()`.

6.304.2.2 u_char [sscop_arg::data](#) []

Definition at line 97 of file `ng_sscop.h`.

6.304.2.3 uint32_t [sscop_arg::sig](#)

Definition at line 95 of file `ng_sscop.h`.

Referenced by `sscfu_send_lower()`, and `sscop_send_upper()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_sscop.h`

6.305 sscop_marg Struct Reference

```
#include <ng_sscop.h>
```

Data Fields

- `uint32_t sig`
- `u_char data []`

6.305.1 Detailed Description

Definition at line 100 of file `ng_sscop.h`.

6.305.2 Field Documentation

6.305.2.1 `u_char sscop_marg::data[]`

Definition at line 102 of file `ng_sscop.h`.

6.305.2.2 `uint32_t sscop_marg::sig`

Definition at line 101 of file `ng_sscop.h`.

Referenced by `sscop_send_manage()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_sscop.h`

6.306 sscop_merr Struct Reference

```
#include <ng_sscop.h>
```

Data Fields

- [uint32_t sig](#)
- [uint32_t err](#)
- [uint32_t cnt](#)

6.306.1 Detailed Description

Definition at line 104 of file `ng_sscop.h`.

6.306.2 Field Documentation

6.306.2.1 `uint32_t sscop_merr::cnt`

Definition at line 107 of file `ng_sscop.h`.

Referenced by `sscop_send_manage()`.

6.306.2.2 `uint32_t sscop_merr::err`

Definition at line 106 of file `ng_sscop.h`.

Referenced by `sscop_send_manage()`.

6.306.2.3 `uint32_t sscop_merr::sig`

Definition at line 105 of file `ng_sscop.h`.

Referenced by `sscop_send_manage()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_sscop.h](#)

6.307 stats Struct Reference

Data Fields

- [uint64_t in_packets](#)
- [uint64_t out_packets](#)
- [uint64_t aa_signals](#)
- [uint64_t errors](#)
- [uint64_t data_delivered](#)
- [uint64_t aa_dropped](#)
- [uint64_t maa_dropped](#)
- [uint64_t maa_signals](#)
- [uint64_t in_dropped](#)
- [uint64_t out_dropped](#)

6.307.1 Detailed Description

Definition at line 74 of file `ng_sscoop.c`.

6.307.2 Field Documentation

6.307.2.1 [uint64_t stats::aa_dropped](#)

Definition at line 80 of file `ng_sscoop.c`.

Referenced by `sscoop_send_upper()`, and `text_status()`.

6.307.2.2 [uint64_t stats::aa_signals](#)

Definition at line 77 of file `ng_sscoop.c`.

Referenced by `sscoop_send_upper()`, and `text_status()`.

6.307.2.3 [uint64_t stats::data_delivered](#)

Definition at line 79 of file `ng_sscoop.c`.

Referenced by `sscoop_send_upper()`, and `text_status()`.

6.307.2.4 [uint64_t stats::errors](#)

Definition at line 78 of file `ng_sscoop.c`.

Referenced by `sscoop_send_manage()`, and `text_status()`.

6.307.2.5 [uint64_t stats::in_dropped](#)

Definition at line 83 of file `ng_sscoop.c`.

Referenced by `ng_sscoop_rcvlower()`, and `text_status()`.

6.307.2.6 [uint64_t stats::in_packets](#)

Definition at line 75 of file `ng_sscop.c`.

Referenced by `ng_sscop_revlower()`, and `text_status()`.

6.307.2.7 [uint64_t stats::maa_dropped](#)

Definition at line 81 of file `ng_sscop.c`.

Referenced by `sscop_send_manage()`, and `text_status()`.

6.307.2.8 [uint64_t stats::maa_signals](#)

Definition at line 82 of file `ng_sscop.c`.

Referenced by `sscop_send_manage()`, and `text_status()`.

6.307.2.9 [uint64_t stats::out_dropped](#)

Definition at line 84 of file `ng_sscop.c`.

Referenced by `sscop_send_lower()`, and `text_status()`.

6.307.2.10 [uint64_t stats::out_packets](#)

Definition at line 76 of file `ng_sscop.c`.

Referenced by `sscop_send_lower()`, and `text_status()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/sscop/ng_sscop.c](#)

6.308 typeinfo Struct Reference

```
#include <ng_message.h>
```

Data Fields

- char [type_name](#) [NG_TYPESIZ]
- u_int32_t [numnodes](#)

6.308.1 Detailed Description

Definition at line 290 of file [ng_message.h](#).

6.308.2 Field Documentation

6.308.2.1 u_int32_t [typeinfo::numnodes](#)

Definition at line 292 of file [ng_message.h](#).

Referenced by [ng_generic_msg\(\)](#).

6.308.2.2 char [typeinfo::type_name](#)[NG_TYPESIZ]

Definition at line 291 of file [ng_message.h](#).

Referenced by [ng_generic_msg\(\)](#).

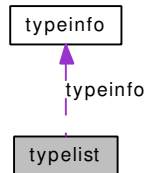
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_message.h](#)

6.309 typelist Struct Reference

```
#include <ng_message.h>
```

Collaboration diagram for typelist:



Data Fields

- `u_int32_t numtypes`
- `typeinfo typeinfo []`

6.309.1 Detailed Description

Definition at line 302 of file `ng_message.h`.

6.309.2 Field Documentation

6.309.2.1 `u_int32_t typelist::numtypes`

Definition at line 303 of file `ng_message.h`.

6.309.2.2 `struct typeinfo typelist::typeinfo[]`

Definition at line 304 of file `ng_message.h`.

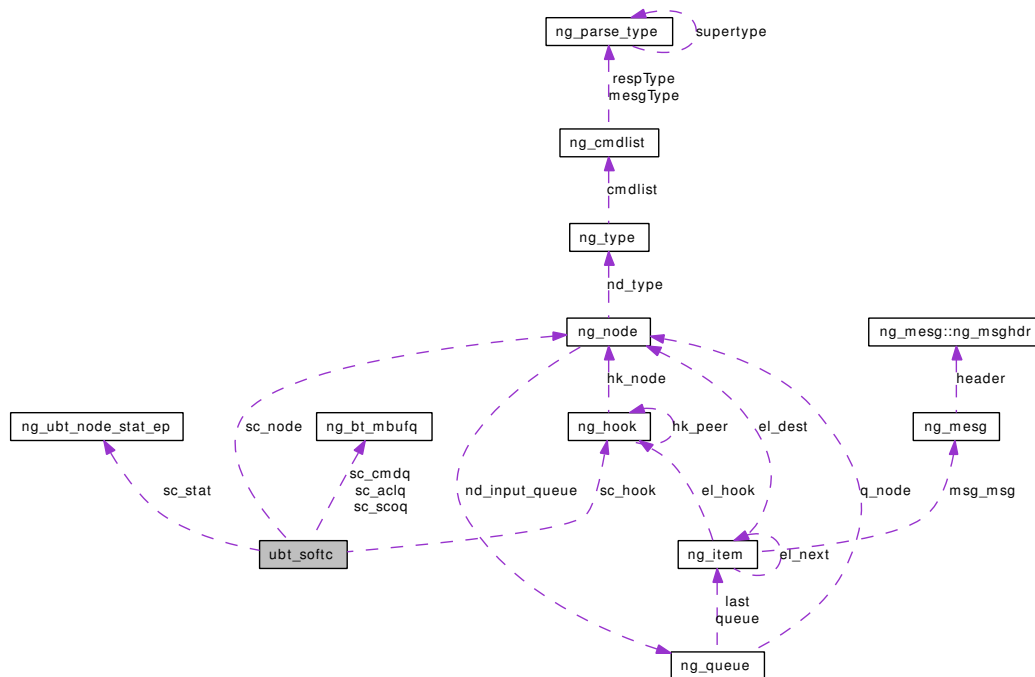
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/ng_message.h`

6.310 ubt_softc Struct Reference

```
#include <ng_ubt_var.h>
```

Collaboration diagram for ubt_softc:



Data Fields

- [ng_ubt_node_debug_ep](#) `sc_debug`
- `u_int32_t` `sc_flags`
- [ng_ubt_node_stat_ep](#) `sc_stat`
- `device_t` `sc_dev`
- `usb_device_handle` `sc_udev`
- `usb_interface_handle` `sc_iface0`
- `usb_interface_handle` `sc_iface1`
- `int` `sc_intr_ep`
- `usb_pipe_handle` `sc_intr_pipe`
- `usb_xfer_handle` `sc_intr_xfer`
- `mbuf *` `sc_intr_buffer`
- `usb_xfer_handle` `sc_ctrl_xfer`
- `void *` `sc_ctrl_buffer`
- [ng_bt_mbufq](#) `sc_cmdq`
- `int` `sc_bulk_in_ep`
- `usb_pipe_handle` `sc_bulk_in_pipe`
- `usb_xfer_handle` `sc_bulk_in_xfer`
- `mbuf *` `sc_bulk_in_buffer`
- `int` `sc_bulk_out_ep`
- `usb_pipe_handle` `sc_bulk_out_pipe`

- [usb_d_xfer_handle](#) [sc_bulk_out_xfer](#)
- [void *](#) [sc_bulk_out_buffer](#)
- [ng_bt_mbufq](#) [sc_aclq](#)
- [int](#) [sc_isoc_in_ep](#)
- [usb_d_pipe_handle](#) [sc_isoc_in_pipe](#)
- [usb_d_xfer_handle](#) [sc_isoc_in_xfer](#)
- [void *](#) [sc_isoc_in_buffer](#)
- [u_int16_t *](#) [sc_isoc_in_frlen](#)
- [int](#) [sc_isoc_out_ep](#)
- [usb_d_pipe_handle](#) [sc_isoc_out_pipe](#)
- [usb_d_xfer_handle](#) [sc_isoc_out_xfer](#)
- [void *](#) [sc_isoc_out_buffer](#)
- [u_int16_t *](#) [sc_isoc_out_frlen](#)
- [ng_bt_mbufq](#) [sc_scoq](#)
- [int](#) [sc_isoc_size](#)
- [u_int32_t](#) [sc_isoc_nframes](#)
- [node_p](#) [sc_node](#)
- [hook_p](#) [sc_hook](#)

6.310.1 Detailed Description

Definition at line 58 of file `ng_ubt_var.h`.

6.310.2 Field Documentation

6.310.2.1 `struct ng_bt_mbufq ubt_softc::sc_aclq`

Definition at line 115 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvdata()`, `ng_ubt_rcvmsg()`, `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, and `ubt_reset()`.

6.310.2.2 `struct mbuf* ubt_softc::sc_bulk_in_buffer`

Definition at line 108 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_complete2()`, and `ubt_bulk_in_start()`.

6.310.2.3 `int ubt_softc::sc_bulk_in_ep`

Definition at line 105 of file `ng_ubt_var.h`.

6.310.2.4 `usb_d_pipe_handle ubt_softc::sc_bulk_in_pipe`

Definition at line 106 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_complete2()`, `ubt_bulk_in_start()`, and `ubt_reset()`.

6.310.2.5 `usb_d_xfer_handle` [ubt_softc::sc_bulk_in_xfer](#)

Definition at line 107 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_start()`.

6.310.2.6 `void*` [ubt_softc::sc_bulk_out_buffer](#)

Definition at line 114 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_out_start()`.

6.310.2.7 `int` [ubt_softc::sc_bulk_out_ep](#)

Definition at line 111 of file `ng_ubt_var.h`.

6.310.2.8 `usb_d_pipe_handle` [ubt_softc::sc_bulk_out_pipe](#)

Definition at line 112 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, and `ubt_reset()`.

6.310.2.9 `usb_d_xfer_handle` [ubt_softc::sc_bulk_out_xfer](#)

Definition at line 113 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_out_start()`.

6.310.2.10 `struct ng_bt_mbufq` [ubt_softc::sc_cmdq](#)

Definition at line 100 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvdata()`, `ng_ubt_rcvmsg()`, `ubt_request_complete2()`, `ubt_request_start()`, and `ubt_reset()`.

6.310.2.11 `void*` [ubt_softc::sc_ctrl_buffer](#)

Definition at line 99 of file `ng_ubt_var.h`.

Referenced by `ubt_request_start()`.

6.310.2.12 `usb_d_xfer_handle` [ubt_softc::sc_ctrl_xfer](#)

Definition at line 98 of file `ng_ubt_var.h`.

Referenced by `ubt_request_start()`.

6.310.2.13 `ng_ubt_node_debug_ep` [ubt_softc::sc_debug](#)

Definition at line 60 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvmsg()`.

6.310.2.14 `device_t ubt_softc::sc_dev`

Definition at line 85 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_connect()`, `ng_ubt_rcvdata()`, `ng_ubt_shutdown()`, `ubt_bulk_in_complete2()`, `ubt_bulk_in_start()`, `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, `ubt_intr_complete2()`, `ubt_intr_start()`, `ubt_isoc_in_complete2()`, `ubt_isoc_in_start()`, `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, `ubt_request_complete2()`, and `ubt_request_start()`.

6.310.2.15 `u_int32_t ubt_softc::sc_flags`

Definition at line 61 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvdata()`, `ng_ubt_rcvmsg()`, `ubt_bulk_in_complete2()`, `ubt_bulk_in_start()`, `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, `ubt_intr_complete2()`, `ubt_intr_start()`, `ubt_isoc_in_complete2()`, `ubt_isoc_in_start()`, `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, `ubt_request_complete2()`, and `ubt_request_start()`.

6.310.2.16 `hook_p ubt_softc::sc_hook`

Definition at line 141 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_connect()`, `ng_ubt_disconnect()`, `ng_ubt_newhook()`, `ng_ubt_rcvdata()`, `ng_ubt_rcvmsg()`, `ubt_bulk_in_complete2()`, `ubt_intr_complete2()`, and `ubt_isoc_in_complete2()`.

6.310.2.17 `usbd_interface_handle ubt_softc::sc_iface0`

Definition at line 88 of file `ng_ubt_var.h`.

6.310.2.18 `usbd_interface_handle ubt_softc::sc_iface1`

Definition at line 89 of file `ng_ubt_var.h`.

6.310.2.19 `struct mbuf* ubt_softc::sc_intr_buffer`

Definition at line 95 of file `ng_ubt_var.h`.

Referenced by `ubt_intr_complete2()`, and `ubt_intr_start()`.

6.310.2.20 `int ubt_softc::sc_intr_ep`

Definition at line 92 of file `ng_ubt_var.h`.

6.310.2.21 `usbd_pipe_handle ubt_softc::sc_intr_pipe`

Definition at line 93 of file `ng_ubt_var.h`.

Referenced by `ubt_intr_complete2()`, `ubt_intr_start()`, and `ubt_reset()`.

6.310.2.22 `usb_d_xfer_handle` [ubt_softc::sc_intr_xfer](#)

Definition at line 94 of file `ng_ubt_var.h`.

Referenced by `ubt_intr_start()`.

6.310.2.23 `void*` [ubt_softc::sc_isoc_in_buffer](#)

Definition at line 123 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_complete2()`.

6.310.2.24 `int` [ubt_softc::sc_isoc_in_ep](#)

Definition at line 120 of file `ng_ubt_var.h`.

6.310.2.25 `u_int16_t*` [ubt_softc::sc_isoc_in_frlen](#)

Definition at line 124 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_complete2()`, and `ubt_isoc_in_start()`.

6.310.2.26 `usb_d_pipe_handle` [ubt_softc::sc_isoc_in_pipe](#)

Definition at line 121 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_complete2()`, `ubt_isoc_in_start()`, and `ubt_reset()`.

6.310.2.27 `usb_d_xfer_handle` [ubt_softc::sc_isoc_in_xfer](#)

Definition at line 122 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_start()`.

6.310.2.28 `u_int32_t` [ubt_softc::sc_isoc_nframes](#)

Definition at line 135 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_complete2()`, `ubt_isoc_in_start()`, and `ubt_isoc_out_start()`.

6.310.2.29 `void*` [ubt_softc::sc_isoc_out_buffer](#)

Definition at line 130 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_out_start()`.

6.310.2.30 `int` [ubt_softc::sc_isoc_out_ep](#)

Definition at line 127 of file `ng_ubt_var.h`.

6.310.2.31 `u_int16_t* ubt_softc::sc_isoc_out_frlen`

Definition at line 131 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_out_start()`.

6.310.2.32 `usb_pipe_handle ubt_softc::sc_isoc_out_pipe`

Definition at line 128 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, and `ubt_reset()`.

6.310.2.33 `usb_xfer_handle ubt_softc::sc_isoc_out_xfer`

Definition at line 129 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_out_start()`.

6.310.2.34 `int ubt_softc::sc_isoc_size`

Definition at line 134 of file `ng_ubt_var.h`.

Referenced by `ubt_isoc_in_complete2()`, `ubt_isoc_in_start()`, and `ubt_isoc_out_start()`.

6.310.2.35 `node_p ubt_softc::sc_node`

Definition at line 140 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_shutdown()`, `ubt_bulk_in_start()`, `ubt_bulk_out_start()`, `ubt_intr_start()`, `ubt_isoc_in_start()`, `ubt_isoc_out_start()`, and `ubt_request_start()`.

6.310.2.36 `struct ng_bt_mbufq ubt_softc::sc_scoq`

Definition at line 132 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvdata()`, `ng_ubt_rcvmsg()`, `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, and `ubt_reset()`.

6.310.2.37 `ng_ubt_node_stat_ep ubt_softc::sc_stat`

Definition at line 75 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvmsg()`, `ubt_bulk_in_complete2()`, `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, `ubt_intr_complete2()`, `ubt_isoc_in_complete2()`, `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, `ubt_request_complete2()`, and `ubt_request_start()`.

6.310.2.38 `usb_device_handle ubt_softc::sc_udev`

Definition at line 86 of file `ng_ubt_var.h`.

Referenced by `ubt_request_start()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt_var.h](#)

6.311 ubtbcmfw_softc Struct Reference

Data Fields

- USBBASEDEVICE [sc_dev](#)
- usbd_device_handle [sc_udev](#)
- cdev * [sc_ctrl_dev](#)
- cdev * [sc_intr_in_dev](#)
- cdev * [sc_bulk_out_dev](#)
- usbd_pipe_handle [sc_intr_in_pipe](#)
- usbd_pipe_handle [sc_bulk_out_pipe](#)
- int [sc_flags](#)
- int [sc_refcnt](#)
- int [sc_dying](#)

6.311.1 Detailed Description

Definition at line 70 of file ubtbcmfw.c.

6.311.2 Field Documentation

6.311.2.1 struct cdev* [ubtbcmfw_softc::sc_bulk_out_dev](#)

Definition at line 75 of file ubtbcmfw.c.

6.311.2.2 usbd_pipe_handle [ubtbcmfw_softc::sc_bulk_out_pipe](#)

Definition at line 77 of file ubtbcmfw.c.

Referenced by [ubtbcmfw_close\(\)](#), [ubtbcmfw_open\(\)](#), [ubtbcmfw_poll\(\)](#), and [ubtbcmfw_write\(\)](#).

6.311.2.3 struct cdev* [ubtbcmfw_softc::sc_ctrl_dev](#)

Definition at line 73 of file ubtbcmfw.c.

6.311.2.4 USBBASEDEVICE [ubtbcmfw_softc::sc_dev](#)

Definition at line 71 of file ubtbcmfw.c.

Referenced by [ubtbcmfw_ioctl\(\)](#), [ubtbcmfw_read\(\)](#), and [ubtbcmfw_write\(\)](#).

6.311.2.5 int [ubtbcmfw_softc::sc_dying](#)

Definition at line 83 of file ubtbcmfw.c.

Referenced by [ubtbcmfw_ioctl\(\)](#), [ubtbcmfw_open\(\)](#), [ubtbcmfw_read\(\)](#), and [ubtbcmfw_write\(\)](#).

6.311.2.6 int ubtbcmfw_softc::sc_flags

Definition at line 78 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), and ubtbcmfw_open().

6.311.2.7 struct cdev* ubtbcmfw_softc::sc_intr_in_dev

Definition at line 74 of file ubtbcmfw.c.

6.311.2.8 usbd_pipe_handle ubtbcmfw_softc::sc_intr_in_pipe

Definition at line 76 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), ubtbcmfw_open(), ubtbcmfw_poll(), and ubtbcmfw_read().

6.311.2.9 int ubtbcmfw_softc::sc_refcnt

Definition at line 82 of file ubtbcmfw.c.

Referenced by ubtbcmfw_ioctl(), ubtbcmfw_read(), and ubtbcmfw_write().

6.311.2.10 usbd_device_handle ubtbcmfw_softc::sc_udev

Definition at line 72 of file ubtbcmfw.c.

Referenced by ubtbcmfw_ioctl(), ubtbcmfw_read(), and ubtbcmfw_write().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ubtbcmfw.c](#)

6.312 uni_arg Struct Reference

```
#include <ng_uni.h>
```

Data Fields

- uint32_t [sig](#)
- uint32_t [cookie](#)
- u_char [data](#) []

6.312.1 Detailed Description

Definition at line 113 of file `ng_uni.h`.

6.312.2 Field Documentation

6.312.2.1 uint32_t [uni_arg::cookie](#)

Definition at line 115 of file `ng_uni.h`.

Referenced by `ng_ccatm_send_uni()`, `ng_ccatm_send_uni_glob()`, and `uni_uni_output()`.

6.312.2.2 u_char [uni_arg::data](#) []

Definition at line 116 of file `ng_uni.h`.

6.312.2.3 uint32_t [uni_arg::sig](#)

Definition at line 114 of file `ng_uni.h`.

Referenced by `ng_ccatm_send_uni()`, `ng_ccatm_send_uni_glob()`, and `uni_uni_output()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/ng_uni.h](#)

6.313 uni_timer Struct Reference

```
#include <ng_uni_cust.h>
```

Data Fields

- callout [c](#)

6.313.1 Detailed Description

Definition at line 81 of file `ng_uni_cust.h`.

6.313.2 Field Documentation

6.313.2.1 struct callout [uni_timer::c](#)

Definition at line 82 of file `ng_uni_cust.h`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/uni/ng_uni_cust.h`

6.314 unimem_debug Struct Reference

Data Fields

- const char * [file](#)
- u_int [lno](#)

6.314.1 Detailed Description

Definition at line 765 of file `ng_uni.c`.

6.314.2 Field Documentation

6.314.2.1 const char* [unimem_debug::file](#)

Definition at line 766 of file `ng_uni.c`.

Referenced by `ng_uni_free()`.

6.314.2.2 u_int [unimem_debug::lno](#)

Definition at line 767 of file `ng_uni.c`.

Referenced by `ng_uni_free()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/uni/ng_uni.c`

6.315 uniq Union Reference

Data Fields

- char [bytes](#) [sizeof(void *)]
- void * [pointer](#)

6.315.1 Detailed Description

Definition at line 258 of file `ng_pppoe.c`.

6.315.2 Field Documentation

6.315.2.1 char [uniq::bytes](#)[sizeof(void *)]

Definition at line 259 of file `ng_pppoe.c`.

Referenced by `pppoe_finduniq()`.

6.315.2.2 void* [uniq::pointer](#)

Definition at line 260 of file `ng_pppoe.c`.

Referenced by `pppoe_finduniq()`.

The documentation for this union was generated from the following file:

- [/usr/src/sys/netgraph/ng_pppoe.c](#)

6.316 vatmpif_header Struct Reference

Data Fields

- [uint32_t cellhdr](#)
- [int32_t seq](#)
- [uint64_t cookie](#)
- [uint8_t aal](#)
- [uint8_t __pad](#) [3]

6.316.1 Detailed Description

Definition at line 83 of file `ng_atmpif.c`.

6.316.2 Field Documentation

6.316.2.1 [uint8_t vatmpif_header::__pad](#)[3]

Definition at line 97 of file `ng_atmpif.c`.

6.316.2.2 [uint8_t vatmpif_header::aal](#)

Definition at line 96 of file `ng_atmpif.c`.

6.316.2.3 [uint32_t vatmpif_header::cellhdr](#)

Definition at line 87 of file `ng_atmpif.c`.

6.316.2.4 [uint64_t vatmpif_header::cookie](#)

Definition at line 95 of file `ng_atmpif.c`.

6.316.2.5 [int32_t vatmpif_header::seq](#)

Definition at line 94 of file `ng_atmpif.c`.

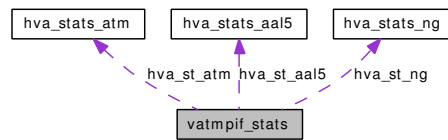
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/atmpif/ng_atmpif.c`

6.317 vatmpif_stats Struct Reference

```
#include <ng_atmpif.h>
```

Collaboration diagram for vatmpif_stats:



Data Fields

- [Hva_Stats_ng](#) [hva_st_ng](#)
- [Hva_Stats_atm](#) [hva_st_atm](#)
- [Hva_Stats_aal5](#) [hva_st_aal5](#)

6.317.1 Detailed Description

Definition at line 137 of file `ng_atmpif.h`.

6.317.2 Field Documentation

6.317.2.1 [Hva_Stats_aal5](#) `vatmpif_stats::hva_st_aal5`

Definition at line 140 of file `ng_atmpif.h`.

6.317.2.2 [Hva_Stats_atm](#) `vatmpif_stats::hva_st_atm`

Definition at line 139 of file `ng_atmpif.h`.

6.317.2.3 [Hva_Stats_ng](#) `vatmpif_stats::hva_st_ng`

Definition at line 138 of file `ng_atmpif.h`.

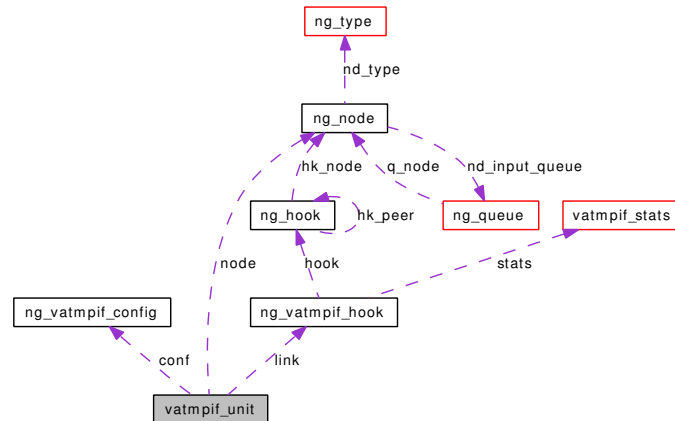
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/ng_atmpif.h`

6.318 vatmpif_unit Struct Reference

```
#include <ng_atmpif_var.h>
```

Collaboration diagram for vatmpif_unit:



Data Fields

- Cmn_unit `vu_cmn`
- `node_p` `node`
- `ng_vatmpif_hook` * `link`
- `ng_vatmpif_config` `conf`

6.318.1 Detailed Description

Definition at line 101 of file `ng_atmpif_var.h`.

6.318.2 Field Documentation

6.318.2.1 struct `ng_vatmpif_config` `vatmpif_unit::conf`

Definition at line 105 of file `ng_atmpif_var.h`.

Referenced by `vatmpif_harp_attach()`.

6.318.2.2 struct `ng_vatmpif_hook`* `vatmpif_unit::link`

Definition at line 104 of file `ng_atmpif_var.h`.

6.318.2.3 `node_p` `vatmpif_unit::node`

Definition at line 103 of file `ng_atmpif_var.h`.

6.318.2.4 Cmn_unit [vatmpif_unit::vu_cmn](#)

Definition at line 102 of file [ng_atmpif_var.h](#).

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h](#)

6.319 vatmpif_vcc Struct Reference

```
#include <ng_atmpif_var.h>
```

Data Fields

- [Cmn_vcc vv_cmn](#)
- [Vatmpif_aal vv_aal](#)
- [Vatmpif_traffic vv_traffic](#)
- [Vatmpif_traffic_type vv_traffic_type](#)

6.319.1 Detailed Description

Definition at line 67 of file `ng_atmpif_var.h`.

6.319.2 Field Documentation

6.319.2.1 [Vatmpif_aal vatmpif_vcc::vv_aal](#)

Definition at line 69 of file `ng_atmpif_var.h`.

Referenced by `vatmpif_harp_openvcc()`.

6.319.2.2 [Cmn_vcc vatmpif_vcc::vv_cmn](#)

Definition at line 68 of file `ng_atmpif_var.h`.

6.319.2.3 [Vatmpif_traffic vatmpif_vcc::vv_traffic](#)

Definition at line 70 of file `ng_atmpif_var.h`.

Referenced by `vatmpif_harp_closevcc()`, and `vatmpif_harp_openvcc()`.

6.319.2.4 [Vatmpif_traffic_type vatmpif_vcc::vv_traffic_type](#)

Definition at line 71 of file `ng_atmpif_var.h`.

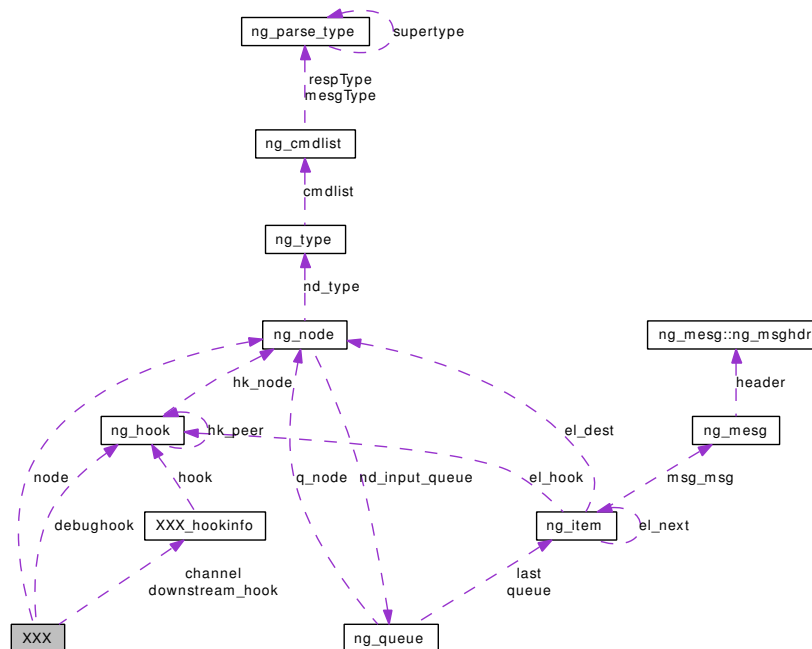
Referenced by `vatmpif_harp_closevcc()`, and `vatmpif_harp_openvcc()`.

The documentation for this struct was generated from the following file:

- `/usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h`

6.320 XXX Struct Reference

Collaboration diagram for XXX:



Data Fields

- [XXX_hookinfo channel](#) [XXX_NUM_DLCIS]
- [XXX_hookinfo downstream_hook](#)
- [node_p node](#)
- [hook_p debughook](#)
- [u_int packets_in](#)
- [u_int packets_out](#)
- [u_int32_t flags](#)

6.320.1 Detailed Description

Definition at line 130 of file ng_sample.c.

6.320.2 Field Documentation

6.320.2.1 struct [XXX_hookinfo](#) XXX::channel[XXX_NUM_DLCIS]

Definition at line 131 of file ng_sample.c.

Referenced by [ng_xxx_newhook\(\)](#), and [ng_xxx_rcvdata\(\)](#).

6.320.2.2 hook_p XXX::debughook

Definition at line 134 of file ng_sample.c.

Referenced by ng_xxx_newhook().

6.320.2.3 struct XXX_hookinfo XXX::downstream_hook

Definition at line 132 of file ng_sample.c.

Referenced by ng_xxx_newhook(), and ng_xxx_rcvdata().

6.320.2.4 u_int32_t XXX::flags

Definition at line 137 of file ng_sample.c.

Referenced by ng_xxx_newhook(), and ng_xxx_rcvmsg().

6.320.2.5 node_p XXX::node

Definition at line 133 of file ng_sample.c.

Referenced by ng_xxx_constructor(), ng_xxx_newhook(), ng_xxx_rcvmsg(), and ng_xxx_shutdown().

6.320.2.6 u_int XXX::packets_in

Definition at line 135 of file ng_sample.c.

Referenced by ng_xxx_rcvdata(), and ng_xxx_rcvmsg().

6.320.2.7 u_int XXX::packets_out

Definition at line 136 of file ng_sample.c.

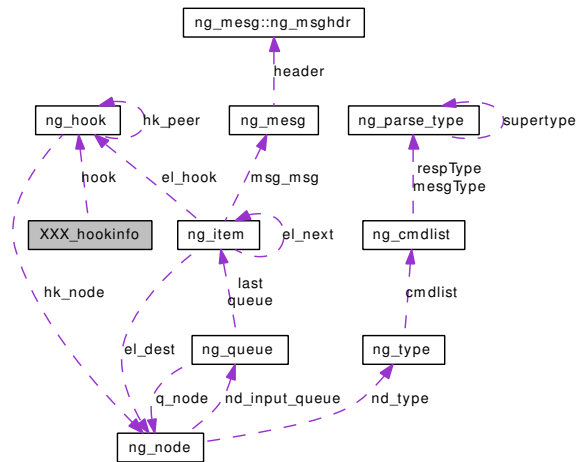
Referenced by ng_xxx_rcvdata(), and ng_xxx_rcvmsg().

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_sample.c](#)

6.321 XXX_hookinfo Struct Reference

Collaboration diagram for XXX_hookinfo:



Data Fields

- `int dlc`
- `int channel`
- `hook_p hook`

6.321.1 Detailed Description

Definition at line 123 of file `ng_sample.c`.

6.321.2 Field Documentation

6.321.2.1 `int XXX_hookinfo::channel`

Definition at line 125 of file `ng_sample.c`.

Referenced by `ng_xxx_rcvdata()`.

6.321.2.2 `int XXX_hookinfo::dlci`

Definition at line 124 of file `ng_sample.c`.

Referenced by `ng_xxx_connect()`, `ng_xxx_newhook()`, and `ng_xxx_rcvdata()`.

6.321.2.3 `hook_p XXX_hookinfo::hook`

Definition at line 126 of file `ng_sample.c`.

Referenced by `ng_xxx_connect()`, `ng_xxx_disconnect()`, `ng_xxx_newhook()`, and `ng_xxx_rcvdata()`.

The documentation for this struct was generated from the following file:

- [/usr/src/sys/netgraph/ng_sample.c](#)

Chapter 7

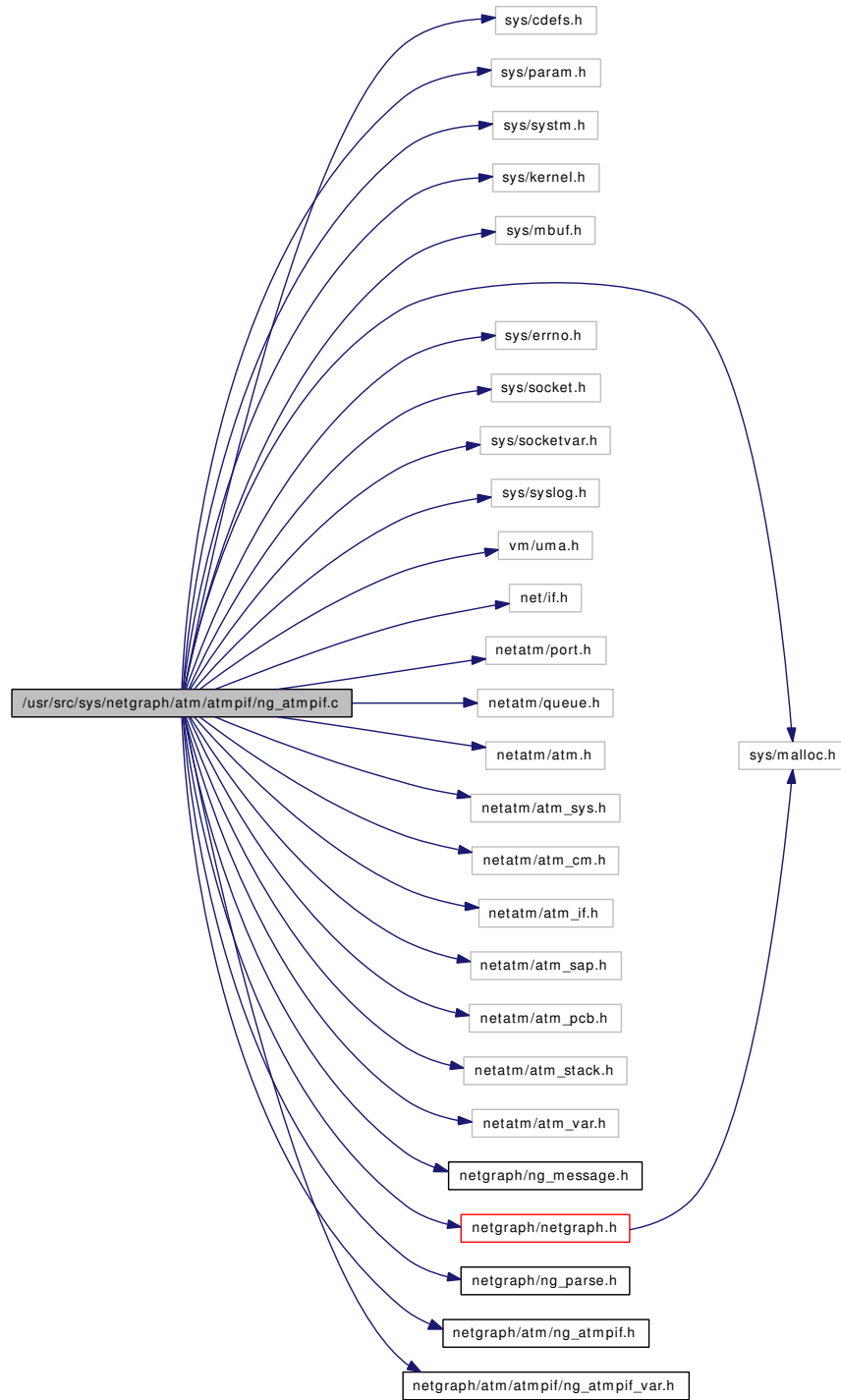
FreeBSD kernel netgraph code File Documentation

7.1 notreviewed.dox File Reference

7.2 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/syslog.h>
#include <vm/uma.h>
#include <net/if.h>
#include <netatm/port.h>
#include <netatm/queue.h>
#include <netatm/atm.h>
#include <netatm/atm_sys.h>
#include <netatm/atm_cm.h>
#include <netatm/atm_if.h>
#include <netatm/atm_sap.h>
#include <netatm/atm_pcb.h>
#include <netatm/atm_stack.h>
#include <netatm/atm_var.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/atm/ng_atmpif.h>
#include <netgraph/atm/atmpif/ng_atmpif_var.h>
```

Include dependency graph for ng_atmpif.c:



Data Structures

- struct [vatmpif_header](#)

Defines

- #define `M_NETGRAPH_ATMPIF` `M_NETGRAPH`

Functions

- `__FBSDID` ("\$FreeBSD: src/sys/netgraph/atm/atmpif/ng_atmpif.c,v 1.3 2005/01/07 01:45:41 imp Exp \$")
- static int `ng_atmpif_mod_event` (module_t, int, void *)
- `NETGRAPH_INIT` (atmpif,&ng_atmpif_typestruct)
- static int `ng_atmpif_constructor` (node_p nodep)
- static int `ng_atmpif_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_atmpif_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_atmpif_disconnect` (hook_p hook)
- static int `ng_atmpif_rmnode` (node_p node)
- static int `ng_atmpif_rcvdata` (hook_p hook, item_p item)
- int `ng_atmpif_transmit` (const priv_p priv, struct mbuf *m, uint8_t vpi, uint16_t vci, uint8_t pt, uint8_t clp, Vatmpif_aal aal)
- static int `ng_macaddr_parse` (const struct ng_parse_type *type, const char *s, int *const off, const u_char *const start, u_char *const buf, int *const buflen)
- static int `ng_macaddr_unparse` (const struct ng_parse_type *type, const u_char *data, int *off, char *cbuf, int cbuflen)

Variables

- static `ng_parse_t` `ng_macaddr_parse`
- static `ng_unparse_t` `ng_macaddr_unparse`
- `ng_parse_type` `ng_mac_addr_type`
- static struct `ng_parse_struct_field` `ng_atmpif_config_type_fields` [] = `NG_ATMPIF_CONFIG_TYPE_INFO`
- static struct `ng_parse_type` `ng_atmpif_config_type`
- static struct `ng_parse_struct_field` `ng_atmpif_link_status_type_fields` [] = `NG_ATMPIF_LINK_STATUS_TYPE_INFO`
- static struct `ng_parse_type` `ng_atmpif_link_status_type`
- static struct `ng_parse_struct_field` `ng_atmpif_stats_type_fields` [] = `NG_ATMPIF_STATS_TYPE_INFO`
- static struct `ng_parse_type` `ng_atmpif_stats_type`
- static struct `ng_cmdlist` `ng_atmpif_cmdlist` []
- `uma_zone_t` `vatmpif_nif_zone`
- `uma_zone_t` `vatmpif_vcc_zone`
- static `ng_constructor_t` `ng_atmpif_constructor`
- static `ng_rcvmsg_t` `ng_atmpif_rcvmsg`
- static `ng_shutdown_t` `ng_atmpif_rmnode`
- static `ng_newhook_t` `ng_atmpif_newhook`
- static `ng_rcvdata_t` `ng_atmpif_rcvdata`
- static `ng_disconnect_t` `ng_atmpif_disconnect`
- static struct `ng_type` `ng_atmpif_typestruct`

7.2.1 Define Documentation

7.2.1.1 #define M_NETGRAPH_ATMPIF M_NETGRAPH

Definition at line 73 of file ng_atmpif.c.

Referenced by ng_atmpif_constructor(), ng_atmpif_disconnect(), ng_atmpif_newhook(), and ng_atmpif_rmnode().

7.2.2 Function Documentation

7.2.2.1 __FBSDID ("\$FreeBSD: src/sys/netgraph/atm/atmpif/ng_atmpif.c, v 1.3 2005/01/07 01:45:41 imp Exp \$")

7.2.2.2 NETGRAPH_INIT (atmpif, & ng_atmpif_typestruct)

7.2.2.3 static int ng_atmpif_constructor (node_p nodep) [static]

Definition at line 225 of file ng_atmpif.c.

References M_NETGRAPH_ATMPIF, and NG_NODE_SET_PRIVATE.

7.2.2.4 static int ng_atmpif_disconnect (hook_p hook) [static]

Definition at line 405 of file ng_atmpif.c.

References ng_vatmpif_hook::hook, M_NETGRAPH_ATMPIF, NG_HOOK_NODE, NG_NODE_PRIVATE, and vatmpif_harp_detach().

Here is the call graph for this function:



7.2.2.5 static int ng_atmpif_mod_event (module_t, int, void *) [static]

Definition at line 670 of file ng_atmpif.c.

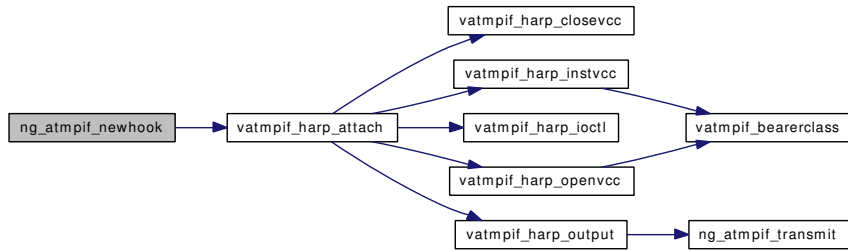
References vatmpif_nif_zone, and vatmpif_vcc_zone.

7.2.2.6 static int ng_atmpif_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 257 of file ng_atmpif.c.

References M_NETGRAPH_ATMPIF, NG_ATMPIF_HOOK_LINK, NG_NODE_PRIVATE, and vatmpif_harp_attach().

Here is the call graph for this function:



7.2.2.7 static int ng_atmpif_rcvdata (hook_p hook, item_p item) [static]

Definition at line 462 of file ng_atmpif.c.

References ng_vatmpif_hook::hook, NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, NGL_GET_M, and vatmpif_harp_recv_drain().

Here is the call graph for this function:



7.2.2.8 static int ng_atmpif_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 299 of file ng_atmpif.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_vatmpif_hook::cur_pcr, ng_mesg::data, ng_mesg::header, ng_vatmpif_hook::InSeq, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGL_GET_MSG, NGM_ATMPIF_CLR_STATS, NGM_ATMPIF_COOKIE, NGM_ATMPIF_GET_CONFIG, NGM_ATMPIF_GET_LINK_STATUS, NGM_ATMPIF_GET_STATS, NGM_ATMPIF_GETCLR_STATS, NGM_ATMPIF_SET_CONFIG, ng_vatmpif_hook::OutSeq, ng_vatmpif_hook::stats, and ng_mesg::ng_msghdr::typecookie.

7.2.2.9 static int ng_atmpif_rmnode (node_p node) [static]

Definition at line 439 of file ng_atmpif.c.

References M_NETGRAPH_ATMPIF, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.2.2.10 int ng_atmpif_transmit (const priv_p priv, struct mbuf * m, uint8_t vpi, uint16_t vci, uint8_t pt, uint8_t clp, Vatmpif_aal aal)

Definition at line 562 of file ng_atmpif.c.

References IS_VATMPIF_DEBUG_PACKET, and NG_SEND_DATA_ONLY.

Referenced by vatmpif_harp_output().

7.2.2.11 `static int ng_macaddr_parse (const struct ng_parse_type * type, const char * s, int *const off, const u_char *const start, u_char *const buf, int *const buflen)` [static]

Definition at line 625 of file `ng_atmpif.c`.

7.2.2.12 `static int ng_macaddr_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbufen)` [static]

Definition at line 652 of file `ng_atmpif.c`.

7.2.3 Variable Documentation

7.2.3.1 `struct ng_cmdlist ng_atmpif_cmdlist[]` [static]

Definition at line 137 of file `ng_atmpif.c`.

7.2.3.2 `struct ng_parse_type ng_atmpif_config_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atmpif_config_type_fields,
}
```

Definition at line 116 of file `ng_atmpif.c`.

7.2.3.3 `struct ng_parse_struct_field ng_atmpif_config_type_fields[] = NG_ATMPIF_CONFIG_TYPE_INFO` [static]

Definition at line 115 of file `ng_atmpif.c`.

7.2.3.4 `ng_constructor_t ng_atmpif_constructor` [static]

Definition at line 190 of file `ng_atmpif.c`.

7.2.3.5 `ng_disconnect_t ng_atmpif_disconnect` [static]

Definition at line 195 of file `ng_atmpif.c`.

7.2.3.6 `struct ng_parse_type ng_atmpif_link_status_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atmpif_link_status_type_fields,
}
```

Definition at line 124 of file `ng_atmpif.c`.

7.2.3.7 `struct ng_parse_struct_field ng_atmpif_link_status_type_fields[] = NG_ATMPIF_LINK_STATUS_TYPE_INFO` [static]

Definition at line 123 of file ng_atmpif.c.

7.2.3.8 `ng_newhook_t ng_atmpif_newhook` [static]

Definition at line 193 of file ng_atmpif.c.

7.2.3.9 `ng_rcvdata_t ng_atmpif_rcvdata` [static]

Definition at line 194 of file ng_atmpif.c.

7.2.3.10 `ng_rcvmsg_t ng_atmpif_rcvmsg` [static]

Definition at line 191 of file ng_atmpif.c.

7.2.3.11 `ng_shutdown_t ng_atmpif_rmnode` [static]

Definition at line 192 of file ng_atmpif.c.

7.2.3.12 `struct ng_parse_type ng_atmpif_stats_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atmpif_stats_type_fields,
}
```

Definition at line 132 of file ng_atmpif.c.

7.2.3.13 `struct ng_parse_struct_field ng_atmpif_stats_type_fields[] = NG_ATMPIF_STATS_TYPE_INFO` [static]

Definition at line 131 of file ng_atmpif.c.

7.2.3.14 `struct ng_type ng_atmpif_tpestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_ATMPIF_NODE_TYPE,
    .mod_event =   ng_atmpif_mod_event,
    .constructor = ng_atmpif_constructor,
    .rcvmsg =      ng_atmpif_rcvmsg,
    .shutdown =    ng_atmpif_rmnode,
    .newhook =     ng_atmpif_newhook,
    .rcvdata =     ng_atmpif_rcvdata,
    .disconnect =  ng_atmpif_disconnect,
```

```
        .cmdlist =      ng_atmpif_cmdlist,  
    }
```

Definition at line 201 of file ng_atmpif.c.

7.2.3.15 struct [ng_parse_type](#) [ng_mac_addr_type](#)

Initial value:

```
{  
    parse: ng_macaddr_parse,  
    unparse: ng_macaddr_unparse,  
}
```

Definition at line 107 of file ng_atmpif.c.

7.2.3.16 [ng_parse_t](#) [ng_macaddr_parse](#) [static]

Definition at line 105 of file ng_atmpif.c.

7.2.3.17 [ng_unparse_t](#) [ng_macaddr_unparse](#) [static]

Definition at line 106 of file ng_atmpif.c.

7.2.3.18 [uma_zone_t](#) [vatmpif_nif_zone](#)

Definition at line 184 of file ng_atmpif.c.

Referenced by [ng_atmpif_mod_event\(\)](#), and [vatmpif_harp_attach\(\)](#).

7.2.3.19 [uma_zone_t](#) [vatmpif_vcc_zone](#)

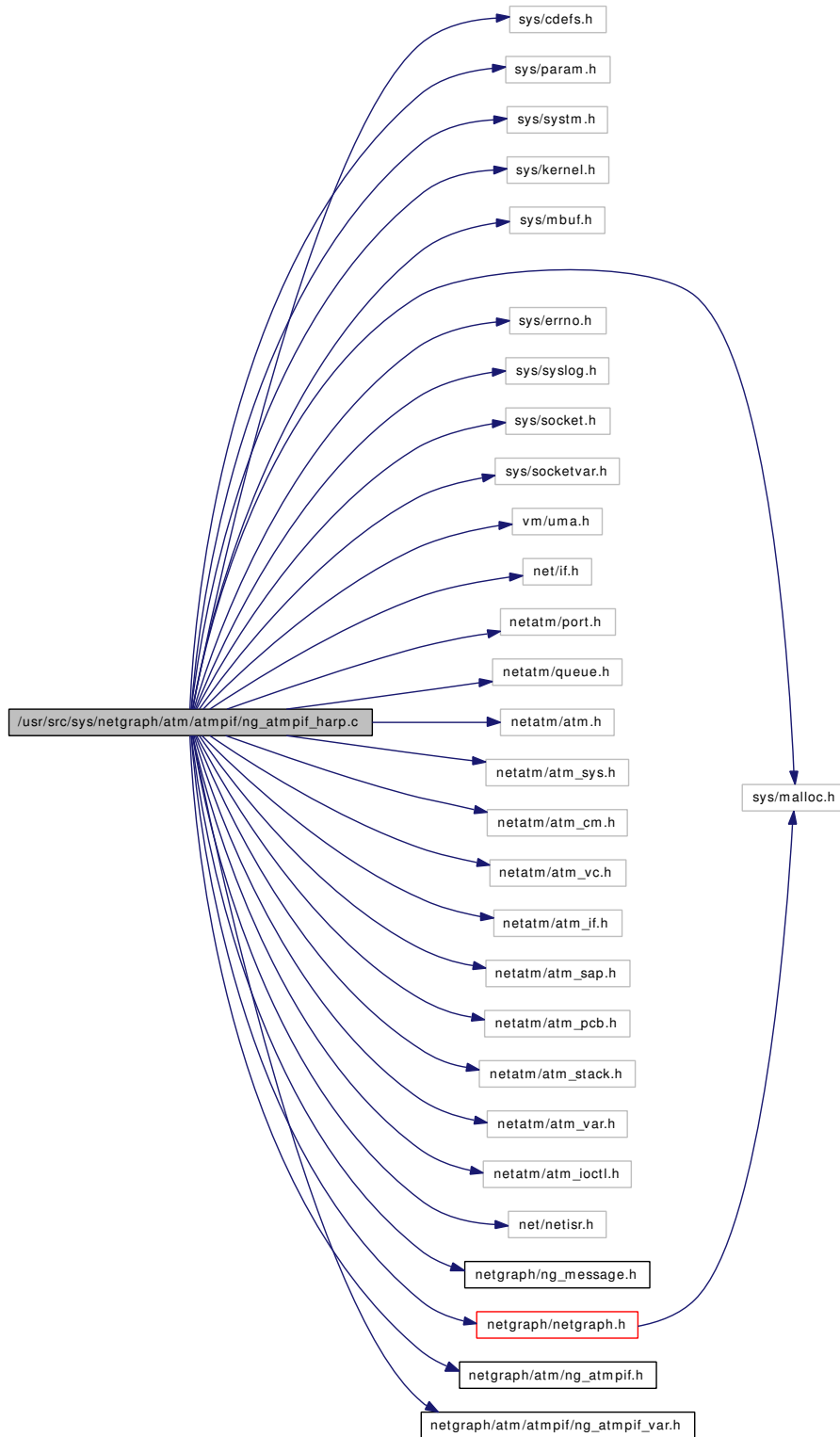
Definition at line 185 of file ng_atmpif.c.

Referenced by [ng_atmpif_mod_event\(\)](#), and [vatmpif_harp_attach\(\)](#).

7.3 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <vm/uma.h>
#include <net/if.h>
#include <netatm/port.h>
#include <netatm/queue.h>
#include <netatm/atm.h>
#include <netatm/atm_sys.h>
#include <netatm/atm_cm.h>
#include <netatm/atm_vc.h>
#include <netatm/atm_if.h>
#include <netatm/atm_sap.h>
#include <netatm/atm_pcb.h>
#include <netatm/atm_stack.h>
#include <netatm/atm_var.h>
#include <netatm/atm_ioctl.h>
#include <net/netisr.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/atm/ng_atmpif.h>
#include <netgraph/atm/atmpif/ng_atmpif_var.h>
```

Include dependency graph for ng_atmpif_harp.c:



Functions

- `__FBSDID` ("FreeBSD: src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c,v 1.5 2005/08/10 06:25:40 obrien Exp \$")
- static int `vatmpif_harp_ioctl` (int code, caddr_t data, caddr_t arg)
- static int `vatmpif_harp_instvcc` (Cmn_unit *cup, Cmn_vcc *cvp)
- static int `vatmpif_harp_openvcc` (Cmn_unit *cup, Cmn_vcc *cvp)
- static int `vatmpif_harp_closevcc` (Cmn_unit *cup, Cmn_vcc *cvp)
- static void `vatmpif_harp_output` (Cmn_unit *cup, Cmn_vcc *cvp, KBuffer *m)
- int `vatmpif_harp_attach` (node_p node)
- int `vatmpif_harp_detach` (node_p node)
- static `Vatmpif_traffic_type` `vatmpif_bearerclass` (struct attr_bearer *bearer)
- static void `vatmpif_harp_recv_stack` (void *tok, KBuffer *m)
- int `vatmpif_harp_recv_drain` (Vatmpif_unit *vup, KBuffer *m, uint8_t vpi, uint16_t vci, uint8_t pt, uint8_t clp, `Vatmpif_aal` aal)

Variables

- static int `vatmpif_nunits` = 0
- static struct stack_defn `vatmpif_svaal5`
- static struct stack_defn `vatmpif_svaal4`
- static struct stack_defn `vatmpif_svaal0`
- static struct stack_defn * `vatmpif_services` = &`vatmpif_svaal0`
- static atm_intr_t `vatmpif_harp_recv_stack`

7.3.1 Function Documentation

7.3.1.1 `__FBSDID` ("FreeBSD: src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c, v 1.5 2005/08/10 06:25:40 obrien Exp \$")

7.3.1.2 static `Vatmpif_traffic_type` `vatmpif_bearerclass` (struct attr_bearer * *bearer*) [static]

Definition at line 392 of file `ng_atmpif_harp.c`.

References `VATMPIF_TRAF_ABR`, `VATMPIF_TRAF_CBR`, `VATMPIF_TRAF_UBR`, and `VATMPIF_TRAF_VBR`.

Referenced by `vatmpif_harp_instvcc()`, and `vatmpif_harp_openvcc()`.

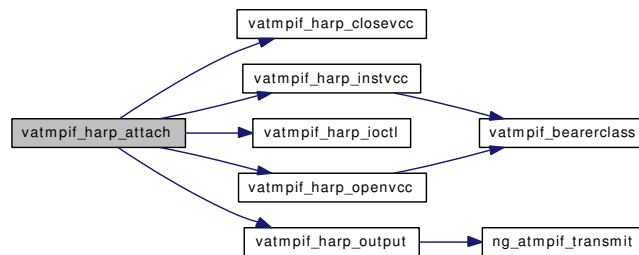
7.3.1.3 int `vatmpif_harp_attach` (node_p *node*)

Definition at line 145 of file `ng_atmpif_harp.c`.

References `vatmpif_unit::conf`, `ng_vatmpif_config::macaddr`, `NG_NODE_PRIVATE`, `NGM_ATMPIF_COOKIE`, `ng_vatmpif_config::pcr`, `VATMPIF_DEV_NAME`, `vatmpif_harp_closevcc()`, `vatmpif_harp_instvcc()`, `vatmpif_harp_ioctl()`, `vatmpif_harp_openvcc()`, `vatmpif_harp_output()`, `VATMPIF_MAX_VCI`, `VATMPIF_MAX_VPI`, `vatmpif_nif_zone`, `vatmpif_nunits`, `vatmpif_services`, and `vatmpif_vcc_zone`.

Referenced by `ng_atmpif_newhook()`.

Here is the call graph for this function:



7.3.1.4 static int vatmpif_harp_closevcc (Cmn_unit * cup, Cmn_vcc * cvp) [static]

Definition at line 602 of file ng_atmpif_harp.c.

References VATMPIF_TRAF_ABR, VATMPIF_TRAF_CBR, VATMPIF_TRAF_UBR, VATMPIF_TRAF_VBR, vatmpif_vcc::vv_traffic, and vatmpif_vcc::vv_traffic_type.

Referenced by vatmpif_harp_attach().

7.3.1.5 int vatmpif_harp_detach (node_p node)

Definition at line 249 of file ng_atmpif_harp.c.

References NG_NODE_PRIVATE, and vatmpif_nunits.

Referenced by ng_atmpif_disconnect().

7.3.1.6 static int vatmpif_harp_instvcc (Cmn_unit * cup, Cmn_vcc * cvp) [static]

Definition at line 435 of file ng_atmpif_harp.c.

References vatmpif_bearerclass(), VATMPIF_TRAF_CBR, VATMPIF_TRAF_UBR, and VATMPIF_TRAF_VBR.

Referenced by vatmpif_harp_attach().

Here is the call graph for this function:



7.3.1.7 static int vatmpif_harp_ioctl (int code, caddr_t data, caddr_t arg) [static]

Definition at line 284 of file ng_atmpif_harp.c.

Referenced by vatmpif_harp_attach().

7.3.1.8 static int vatmpif_harp_openvcc (Cmn_unit * cup, Cmn_vcc * cvp) [static]

Definition at line 520 of file ng_atmpif_harp.c.

References VATMPIF_AAL_0, VATMPIF_AAL_4, VATMPIF_AAL_5, vatmpif_bearerclass(), VATMPIF_TRAF_ABR, VATMPIF_TRAF_CBR, VATMPIF_TRAF_UBR, VATMPIF_TRAF_VBR, vatmpif_vcc::vv_aal, vatmpif_vcc::vv_traffic, and vatmpif_vcc::vv_traffic_type.

Referenced by vatmpif_harp_attach().

Here is the call graph for this function:



7.3.1.9 static void vatmpif_harp_output (Cmn_unit * cup, Cmn_vcc * cvp, KBuffer * m) [static]

Definition at line 653 of file ng_atmpif_harp.c.

References IS_VATMPIF_DEBUG_PACKET, ng_atmpif_transmit(), VATMPIF_AAL_0, VATMPIF_AAL_4, and VATMPIF_AAL_5.

Referenced by vatmpif_harp_attach().

Here is the call graph for this function:



7.3.1.10 int vatmpif_harp_recv_drain (Vatmpif_unit * vup, KBuffer * m, uint8_t vpi, uint16_t vci, uint8_t pt, uint8_t clp, Vatmpif_aal aal)

Definition at line 779 of file ng_atmpif_harp.c.

References IS_VATMPIF_DEBUG_PACKET, VATMPIF_AAL_0, VATMPIF_AAL_4, VATMPIF_AAL_5, and vatmpif_harp_recv_stack.

Referenced by ng_atmpif_rcvdata().

7.3.1.11 static void vatmpif_harp_recv_stack (void * tok, KBuffer * m) [static]

Definition at line 734 of file ng_atmpif_harp.c.

7.3.2 Variable Documentation

7.3.2.1 atm_intr_t vatmpif_harp_recv_stack [static]

Definition at line 129 of file ng_atmpif_harp.c.

Referenced by vatmpif_harp_recv_drain().

7.3.2.2 int vatmpif_nunits = 0 [static]

Definition at line 80 of file ng_atmpif_harp.c.

Referenced by vatmpif_harp_attach(), and vatmpif_harp_detach().

7.3.2.3 struct stack_defn* vatmpif_services = &vatmpif_svaal0 [static]

Definition at line 115 of file ng_atmpif_harp.c.

Referenced by vatmpif_harp_attach().

7.3.2.4 struct stack_defn vatmpif_svaal0 [static]**Initial value:**

```
{
    sd_next: &vatmpif_svaal4,
    sd_sap: SAP_ATM,
    sd_flag: SDF_TERM,
    sd_inst: atm_dev_inst,
    sd_lower: atm_dev_lower,
    sd_upper: NULL,
    sd_toku: 0,
}
```

Definition at line 106 of file ng_atmpif_harp.c.

7.3.2.5 struct stack_defn vatmpif_svaal4 [static]**Initial value:**

```
{
    sd_next: &vatmpif_svaal5,
    sd_sap: SAP_CPCS_AAL3_4,
    sd_flag: SDF_TERM,
    sd_inst: atm_dev_inst,
    sd_lower: atm_dev_lower,
    sd_upper: NULL,
    sd_toku: 0,
}
```

Definition at line 97 of file ng_atmpif_harp.c.

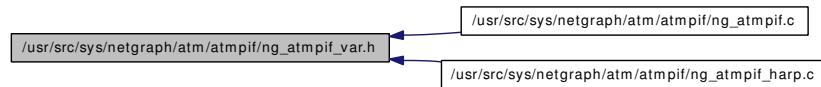
7.3.2.6 struct stack_defn vatmpif_svaal5 [static]**Initial value:**

```
{
    sd_next: NULL,
    sd_sap: SAP_CPCS_AAL5,
    sd_flag: SDF_TERM,
    sd_inst: atm_dev_inst,
    sd_lower: atm_dev_lower,
    sd_upper: NULL,
    sd_toku: 0,
}
```

Definition at line 88 of file ng_atmpif_harp.c.

7.4 /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [vatmpif_vcc](#)
- struct [ng_vatmpif_hook](#)
- struct [vatmpif_unit](#)

Defines

- #define [vv_next](#) vv_cmn.cv_next
- #define [vv_toku](#) vv_cmn.cv_toku
- #define [vv_upper](#) vv_cmn.cv_upper
- #define [vv_connvc](#) vv_cmn.cv_connvc
- #define [vv_state](#) vv_cmn.cv_state
- #define [ng_vatmpif_private](#) vatmpif_unit
- #define [vu_pif](#) vu_cmn.cu_pif
- #define [vu_unit](#) vu_cmn.cu_unit
- #define [vu_flags](#) vu_cmn.cu_flags
- #define [vu_mtu](#) vu_cmn.cu_mtu
- #define [vu_open_vcc](#) vu_cmn.cu_open_vcc
- #define [vu_vcc](#) vu_cmn.cu_vcc
- #define [vu_vcc_zone](#) vu_cmn.cu_vcc_zone
- #define [vu_nif_zone](#) vu_cmn.cu_nif_zone
- #define [vu_ioctl](#) vu_cmn.cu_ioctl
- #define [vu_instvcc](#) vu_cmn.cu_instvcc
- #define [vu_openvcc](#) vu_cmn.cu_openvcc
- #define [vu_closevcc](#) vu_cmn.cu_closevcc
- #define [vu_output](#) vu_cmn.cu_output
- #define [vu_config](#) vu_cmn.cu_config
- #define [vu_softc](#) vu_cmn.cu_softc
- #define [vu_stats](#) link → stats
- #define [vu_cur_pcr](#) link → cur_pcr

Typedefs

- typedef enum [vatmpif_aal](#) Vatmpif_aal
- typedef enum [vatmpif_traffic_type](#) Vatmpif_traffic_type
- typedef t_atm_traffic [Vatmpif_traffic](#)
- typedef [vatmpif_vcc](#) Vatmpif_vcc
- typedef [vatmpif_unit](#) Vatmpif_unit
- typedef ng_vatmpif_private * [priv_p](#)

Enumerations

- enum `vatmpif_aal` { `VATMPIF_AAL_0` = 0, `VATMPIF_AAL_4` = 4, `VATMPIF_AAL_5` = 5 }
- enum `vatmpif_traffic_type` { `VATMPIF_TRAF_CBR` = 0x01, `VATMPIF_TRAF_VBR` = 0x02, `VATMPIF_TRAF_ABR` = 0x03, `VATMPIF_TRAF_UBR` = 0x04 }

Functions

- int `vatmpif_harp_attach` (`node_p` node)
- int `vatmpif_harp_detach` (`node_p` node)
- int `vatmpif_harp_recv_drain` (`Vatmpif_unit` *vup, `KBuffer` *m, `uint8_t` vpi, `uint16_t` vci, `uint8_t` pt, `uint8_t` clp, `Vatmpif_aal` aal)
- int `ng_atmpif_transmit` (const `priv_p` priv, `struct mbuf` *m, `uint8_t` vpi, `uint16_t` vci, `uint8_t` pt, `uint8_t` clp, `Vatmpif_aal` aal)

Variables

- `uma_zone_t` `vatmpif_nif_zone`
- `uma_zone_t` `vatmpif_vcc_zone`

7.4.1 Define Documentation

7.4.1.1 #define `ng_vatmpif_private` `vatmpif_unit`

Definition at line 109 of file `ng_atmpif_var.h`.

7.4.1.2 #define `vu_closevcc` `vu_cmn.cu_closevcc`

Definition at line 123 of file `ng_atmpif_var.h`.

7.4.1.3 #define `vu_config` `vu_cmn.cu_config`

Definition at line 125 of file `ng_atmpif_var.h`.

7.4.1.4 #define `vu_cur_pcr` `link` → `cur_pcr`

Definition at line 129 of file `ng_atmpif_var.h`.

7.4.1.5 #define `vu_flags` `vu_cmn.cu_flags`

Definition at line 114 of file `ng_atmpif_var.h`.

7.4.1.6 #define `vu_instvcc` `vu_cmn.cu_instvcc`

Definition at line 121 of file `ng_atmpif_var.h`.

7.4.1.7 #define vu_ioctl vu_cmn.cu_ioctl

Definition at line 120 of file ng_atmpif_var.h.

7.4.1.8 #define vu_mtu vu_cmn.cu_mtu

Definition at line 115 of file ng_atmpif_var.h.

7.4.1.9 #define vu_nif_zone vu_cmn.cu_nif_zone

Definition at line 119 of file ng_atmpif_var.h.

7.4.1.10 #define vu_open_vcc vu_cmn.cu_open_vcc

Definition at line 116 of file ng_atmpif_var.h.

7.4.1.11 #define vu_openvcc vu_cmn.cu_openvcc

Definition at line 122 of file ng_atmpif_var.h.

7.4.1.12 #define vu_output vu_cmn.cu_output

Definition at line 124 of file ng_atmpif_var.h.

7.4.1.13 #define vu_pif vu_cmn.cu_pif

Definition at line 112 of file ng_atmpif_var.h.

7.4.1.14 #define vu_softc vu_cmn.cu_softc

Definition at line 126 of file ng_atmpif_var.h.

7.4.1.15 #define vu_stats link → [stats](#)

Definition at line 128 of file ng_atmpif_var.h.

7.4.1.16 #define vu_unit vu_cmn.cu_unit

Definition at line 113 of file ng_atmpif_var.h.

7.4.1.17 #define vu_vcc vu_cmn.cu_vcc

Definition at line 117 of file ng_atmpif_var.h.

7.4.1.18 #define vu_vcc_zone vu_cmn.cu_vcc_zone

Definition at line 118 of file ng_atmpif_var.h.

7.4.1.19 #define vv_connvc vv_cmn.cv_connvc

Definition at line 78 of file ng_atmpif_var.h.

7.4.1.20 #define vv_next vv_cmn.cv_next

Definition at line 75 of file ng_atmpif_var.h.

7.4.1.21 #define vv_state vv_cmn.cv_state

Definition at line 79 of file ng_atmpif_var.h.

7.4.1.22 #define vv_toku vv_cmn.cv_toku

Definition at line 76 of file ng_atmpif_var.h.

7.4.1.23 #define vv_upper vv_cmn.cv_upper

Definition at line 77 of file ng_atmpif_var.h.

7.4.2 Typedef Documentation**7.4.2.1 typedef struct ng_vatmpif_private* [priv_p](#)**

Definition at line 110 of file ng_atmpif_var.h.

7.4.2.2 typedef enum [vatmpif_aal](#) [Vatmpif_aal](#)

Definition at line 38 of file ng_atmpif_var.h.

7.4.2.3 typedef struct t_atm_traffic [Vatmpif_traffic](#)

Definition at line 51 of file ng_atmpif_var.h.

7.4.2.4 typedef enum [vatmpif_traffic_type](#) [Vatmpif_traffic_type](#)

Definition at line 49 of file ng_atmpif_var.h.

7.4.2.5 typedef struct [vatmpif_unit](#) [Vatmpif_unit](#)

Definition at line 107 of file ng_atmpif_var.h.

7.4.2.6 typedef struct [vatmpif_vcc](#) [Vatmpif_vcc](#)

Definition at line 73 of file `ng_atmpif_var.h`.

7.4.3 Enumeration Type Documentation

7.4.3.1 enum [vatmpif_aal](#)

Enumerator:

VATMPIF_AAL_0
VATMPIF_AAL_4
VATMPIF_AAL_5

Definition at line 33 of file `ng_atmpif_var.h`.

7.4.3.2 enum [vatmpif_traffic_type](#)

Enumerator:

VATMPIF_TRAF_CBR
VATMPIF_TRAF_VBR
VATMPIF_TRAF_ABR
VATMPIF_TRAF_UBR

Definition at line 43 of file `ng_atmpif_var.h`.

7.4.4 Function Documentation

7.4.4.1 int `ng_atmpif_transmit` (const [priv_p](#) *priv*, struct `mbuf` * *m*, `uint8_t` *vpi*, `uint16_t` *vci*, `uint8_t` *pt*, `uint8_t` *clp*, [Vatmpif_aal](#) *aal*)

Definition at line 562 of file `ng_atmpif.c`.

References `IS_VATMPIF_DEBUG_PACKET`, and `NG_SEND_DATA_ONLY`.

Referenced by `vatmpif_harp_output()`.

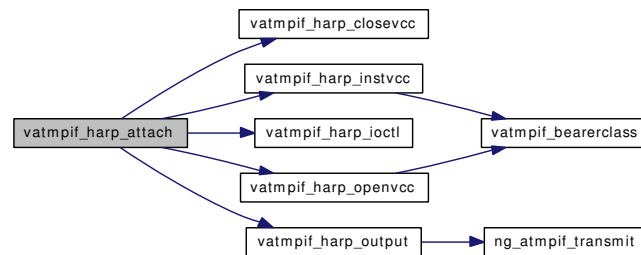
7.4.4.2 int `vatmpif_harp_attach` ([node_p](#) *node*)

Definition at line 145 of file `ng_atmpif_harp.c`.

References `vatmpif_unit::conf`, `ng_vatmpif_config::macaddr`, `NG_NODE_PRIVATE`, `NGM_ATMPIF_COOKIE`, `ng_vatmpif_config::pcr`, `VATMPIF_DEV_NAME`, `vatmpif_harp_closevcc()`, `vatmpif_harp_instvcc()`, `vatmpif_harp_ioctl()`, `vatmpif_harp_openvcc()`, `vatmpif_harp_output()`, `VATMPIF_MAX_VCI`, `VATMPIF_MAX_VPI`, `vatmpif_nif_zone`, `vatmpif_nunits`, `vatmpif_services`, and `vatmpif_vcc_zone`.

Referenced by `ng_atmpif_newhook()`.

Here is the call graph for this function:



7.4.4.3 int vatmpif_harp_detach ([node_p node](#))

Definition at line 249 of file ng_atmpif_harp.c.

References NG_NODE_PRIVATE, and vatmpif_nunits.

Referenced by ng_atmpif_disconnect().

7.4.4.4 int vatmpif_harp_recv_drain ([Vatmpif_unit * vup](#), [KBuffer * m](#), [uint8_t vpi](#), [uint16_t vci](#), [uint8_t pt](#), [uint8_t clp](#), [Vatmpif_aal aal](#))

Definition at line 779 of file ng_atmpif_harp.c.

References IS_VATMPIF_DEBUG_PACKET, VATMPIF_AAL_0, VATMPIF_AAL_4, VATMPIF_AAL_5, and vatmpif_harp_recv_stack.

Referenced by ng_atmpif_rcvdata().

7.4.5 Variable Documentation

7.4.5.1 [uma_zone_t vatmpif_nif_zone](#)

Definition at line 184 of file ng_atmpif.c.

Referenced by ng_atmpif_mod_event(), and vatmpif_harp_attach().

7.4.5.2 [uma_zone_t vatmpif_vcc_zone](#)

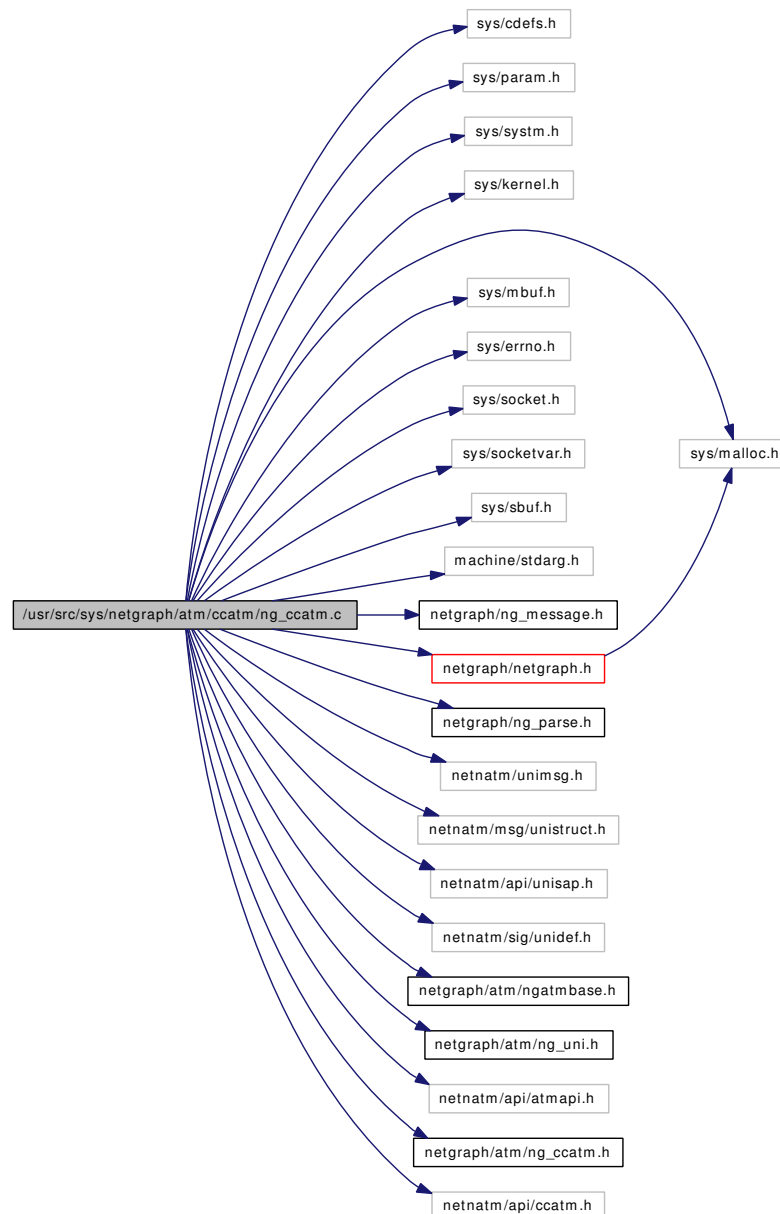
Definition at line 185 of file ng_atmpif.c.

Referenced by ng_atmpif_mod_event(), and vatmpif_harp_attach().

7.5 /usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sbuf.h>
#include <machine/stdarg.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netnatm/unimsg.h>
#include <netnatm/msg/unistruct.h>
#include <netnatm/api/unisap.h>
#include <netnatm/sig/unidef.h>
#include <netgraph/atm/ngatmbase.h>
#include <netgraph/atm/ng_uni.h>
#include <netnatm/api/atmapi.h>
#include <netgraph/atm/ng_ccatm.h>
#include <netnatm/api/ccatm.h>
```

Include dependency graph for ng_ccatm.c:



Data Structures

- struct [ccnode](#)
- struct [cchok](#)

Functions

- [__FBSDID](#) ("\$FreeBSD: src/sys/netgraph/atm/ccatm/ng_ccatm.c,v 1.3 2006/09/30 12:37:43 netchild Exp \$")
- [MODULE_DEPEND](#) (ng_ccatm, ngatmbase, 1, 1, 1)
- [MALLOC_DEFINE](#) (M_NG_CCATM, "ng_ccatm", "netgraph uni api node")

- static int `ng_ccatm_addr_req_array_getlen` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf)
- static int `ng_ccatm_port_array_getlen` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf)
- static int `ng_ccatm_mod_event` (module_t, int, void *)
- `NETGRAPH_INIT` (ccatm,&ng_ccatm_typestruct)
- static void `ng_ccatm_send_user` (struct ccuser *, void *, u_int, void *, size_t)
- static void `ng_ccatm_respond_user` (struct ccuser *, void *, int, u_int, void *, size_t)
- static void `ng_ccatm_send_uni` (struct ccconn *, void *, u_int, u_int, struct uni_msg *)
- static void `ng_ccatm_send_uni_glob` (struct cport *, void *, u_int, u_int, struct uni_msg *)
- static void `ng_ccatm_log` (const char *,...)
- static int `ng_ccatm_constructor` (node_p node)
- static int `ng_ccatm_shutdown` (node_p node)
- static int `ng_ccatm_get_addresses` (node_p node, uint32_t portno, struct `ng_mesg` *msg, struct `ng_mesg` **resp)
- static int `send_dump` (struct ccdata *data, void *uarg, const char *buf)
- static int `ng_ccatm_dump` (node_p node)
- static int `ng_ccatm_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_ccatm_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_ccatm_disconnect` (hook_p hook)
- static int `ng_ccatm_rcvdata` (hook_p hook, item_p item)
- static struct mbuf * `pack_buf` (void *h, size_t hlen, void *t, size_t tlen)
- static int `ng_ccatm_rcvuni` (hook_p hook, item_p item)
- static int `ng_ccatm_rcvmanage` (hook_p hook, item_p item)
- static int `ng_ccatm_rcvdump` (hook_p hook, item_p item)

Variables

- static struct `ng_parse_fixedarray_info` `ng_ccatm_esi_type_info`
- static struct `ng_parse_type` `ng_ccatm_esi_type`
- static struct `ng_parse_struct_field` `ng_ccatm_atm_port_type_info` []
- static struct `ng_parse_type` `ng_ccatm_atm_port_type`
- static struct `ng_parse_struct_field` `ng_ccatm_port_type_info` []
- static struct `ng_parse_type` `ng_ccatm_port_type`
- static struct `ng_parse_fixedarray_info` `ng_ccatm_addr_array_type_info`
- static struct `ng_parse_type` `ng_ccatm_addr_array_type`
- static struct `ng_parse_struct_field` `ng_ccatm_uni_addr_type_info` []
- static struct `ng_parse_type` `ng_ccatm_uni_addr_type`
- static struct `ng_parse_struct_field` `ng_ccatm_addr_req_type_info` []
- static struct `ng_parse_type` `ng_ccatm_addr_req_type`
- static struct `ng_parse_array_info` `ng_ccatm_addr_req_array_type_info`
- static struct `ng_parse_type` `ng_ccatm_addr_req_array_type`
- static struct `ng_parse_struct_field` `ng_ccatm_get_addresses_type_info` []
- static struct `ng_parse_type` `ng_ccatm_get_addresses_type`
- static struct `ng_parse_array_info` `ng_ccatm_port_array_type_info`
- static struct `ng_parse_type` `ng_ccatm_port_array_type`
- static struct `ng_parse_struct_field` `ng_ccatm_portlist_type_info` []
- static struct `ng_parse_type` `ng_ccatm_portlist_type`
- static struct `ng_cmdlist` `ng_ccatm_cmdlist` []
- static `ng_constructor_t` `ng_ccatm_constructor`

- static [ng_rcvmsg_t](#) [ng_ccatm_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_ccatm_shutdown](#)
- static [ng_newhook_t](#) [ng_ccatm_newhook](#)
- static [ng_rcvdata_t](#) [ng_ccatm_rcvdata](#)
- static [ng_disconnect_t](#) [ng_ccatm_disconnect](#)
- static struct [ng_type](#) [ng_ccatm_typestruct](#)
- static [ng_rcvdata_t](#) [ng_ccatm_rcvuni](#)
- static [ng_rcvdata_t](#) [ng_ccatm_rcvdump](#)
- static [ng_rcvdata_t](#) [ng_ccatm_rcvmanage](#)

7.5.1 Function Documentation

7.5.1.1 `__FBSDID ("FreeBSD: src/sys/netgraph/atm/ccatm/ng_ccatm.c, v 1.3 2006/09/30 12:37:43 netchild Exp $")`

7.5.1.2 `MALLOC_DEFINE (M_NG_CCATM, "ng_ccatm", "netgraph uni api node")`

7.5.1.3 `MODULE_DEPEND (ng_ccatm, ngatmbase, 1, 1, 1)`

7.5.1.4 `NETGRAPH_INIT (ccatm, & ng_ccatm_typestruct)`

7.5.1.5 `static int ng_ccatm_addr_req_array_getlen (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]`

Definition at line 123 of file `ng_ccatm.c`.

References `ngm_ccatm_get_addresses::addr`, and `ngm_ccatm_get_addresses::count`.

7.5.1.6 `static int ng_ccatm_constructor (node_p node) [static]`

Definition at line 347 of file `ng_ccatm.c`.

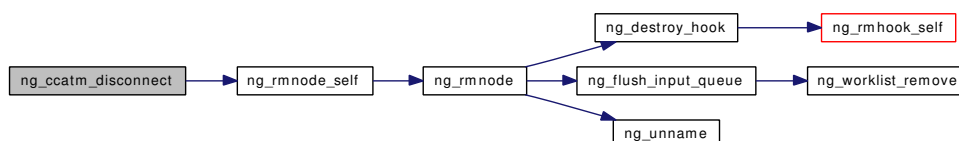
References `NG_NODE_SET_PRIVATE`, and `ccnode::node`.

7.5.1.7 `static int ng_ccatm_disconnect (hook_p hook) [static]`

Definition at line 877 of file `ng_ccatm.c`.

References `ccnode::data`, `ccnode::dump`, `chook::hook`, `ccnode::hook_cnt`, `chook::inst`, `chook::is_`
`uni`, `ccnode::manage`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_HOOK_SET_PRIVATE`, `NG_`
`NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, `ng_rmnode_self()`, `ccnode::node`,
and `chook::node`.

Here is the call graph for this function:



7.5.1.8 static int ng_ccatm_dump (node_p node) [static]

Definition at line 464 of file ng_ccatm.c.

References ccnode::data, ccnode::dump, ccnode::dump_first, ccnode::dump_last, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, ccnode::node, and send_dump().

Referenced by ng_ccatm_rcvmsg().

Here is the call graph for this function:



7.5.1.9 static int ng_ccatm_get_addresses (node_p node, uint32_t portno, struct ng_mesg * msg, struct ng_mesg ** resp) [static]

Definition at line 393 of file ng_ccatm.c.

References ngm_ccatm_addr_req::addr, ngm_ccatm_get_addresses::addr, ngm_ccatm_get_addresses::count, ccnode::data, NG_MKRESPONSE, NG_NODE_PRIVATE, ccnode::node, and ngm_ccatm_addr_req::port.

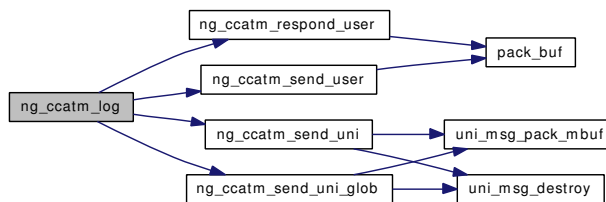
Referenced by ng_ccatm_rcvmsg().

7.5.1.10 static void ng_ccatm_log (const char * fmt, ...) [static]

Definition at line 332 of file ng_ccatm.c.

References ng_ccatm_respond_user(), ng_ccatm_send_user(), ng_ccatm_send_user_glob(), and ng_ccatm_send_user().

Here is the call graph for this function:



7.5.1.11 static int ng_ccatm_mod_event (module_t, int, void *) [static]

Definition at line 1181 of file ng_ccatm.c.

7.5.1.12 static int ng_ccatm_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 794 of file ng_ccatm.c.

References `ccnode::data`, `ccnode::dump`, `cchook::hook`, `ccnode::hook_cnt`, `ccnode::manage`, `ng_ccatm_rcvdump`, `ng_ccatm_rcvmanage`, `ng_ccatm_rcvuni`, `NG_HOOK_FORCE_QUEUE`, `NG_HOOK_NAME`, `NG_HOOK_SET_PRIVATE`, `NG_HOOK_SET_RCVDATA`, `NG_NODE_PRIVATE`, and `ccnode::node`.

7.5.1.13 `static int ng_ccatm_port_array_getlen (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 149 of file `ng_ccatm.c`.

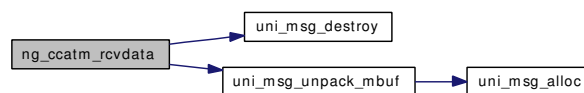
References `ngm_ccatm_portlist::nports`, and `ngm_ccatm_portlist::ports`.

7.5.1.14 `static int ng_ccatm_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 921 of file `ng_ccatm.c`.

References `ccnode::data`, `cchook::hook`, `cchook::inst`, `NG_FREE_ITEM`, `NG_HOOK_PRIVATE`, `NGI_GET_M`, `cchook::node`, `ccatm_op::op`, `uni_msg_destroy()`, and `uni_msg_unpack_mbuf()`.

Here is the call graph for this function:



7.5.1.15 `static int ng_ccatm_rcvdump (hook_p hook, item_p item)` [static]

Definition at line 1160 of file `ng_ccatm.c`.

References `NG_FREE_ITEM`.

7.5.1.16 `static int ng_ccatm_rcvmanage (hook_p hook, item_p item)` [static]

Definition at line 1153 of file `ng_ccatm.c`.

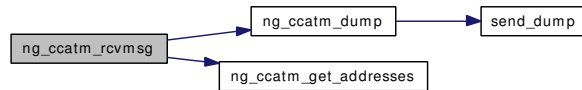
References `NG_FREE_ITEM`.

7.5.1.17 `static int ng_ccatm_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 487 of file `ng_ccatm.c`.

References `ng_msg::ng_msghdr::arglen`, `ng_msg::ng_msghdr::cmd`, `ccnode::data`, `ng_msg::data`, `ccnode::dump`, `ng_msg::header`, `ccnode::hook_cnt`, `ng_ccatm_dump()`, `ng_ccatm_get_addresses()`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_CCATM_ADDRESS_REGISTERED`, `NGM_CCATM_ADDRESS_UNREGISTERED`, `NGM_CCATM_CLEAR`, `NGM_CCATM_COOKIE`, `NGM_CCATM_DUMP`, `NGM_CCATM_GET_ADDRESSES`, `NGM_CCATM_GET_EXSTAT`, `NGM_CCATM_GET_PORT_PARAM`, `NGM_CCATM_GET_PORTLIST`, `NGM_CCATM_GETSTATE`, `NGM_CCATM_RESET`, `NGM_CCATM_SET_PORT_PARAM`, `NGM_CCATM_SETLOG`, `NGM_CCATM_START`, `NGM_CCATM_STOP`, `ccnode::node`, `ngm_ccatm_portlist::ports`, and `ng_msg::ng_msghdr::typecookie`.

Here is the call graph for this function:

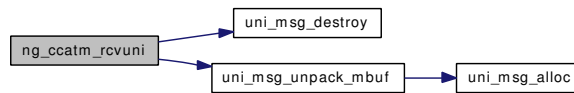


7.5.1.18 static int ng_ccatm_rcvuni (hook_p hook, item_p item) [static]

Definition at line 1055 of file ng_ccatm.c.

References cnode::data, chook::hook, chook::inst, NG_FREE_ITEM, NG_HOOK_PRIVATE, NGI_GET_M, chook::node, uni_msg_destroy(), and uni_msg_unpack_mbuf().

Here is the call graph for this function:



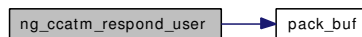
7.5.1.19 static void ng_ccatm_respond_user (struct ccuser *, void *, int, u_int, void *, size_t) [static]

Definition at line 1028 of file ng_ccatm.c.

References chook::hook, NG_SEND_DATA_ONLY, ccatm_op::op, and pack_buf().

Referenced by ng_ccatm_log().

Here is the call graph for this function:



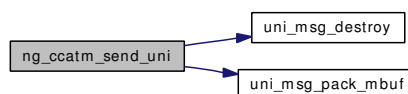
7.5.1.20 static void ng_ccatm_send_uni (struct cconn *, void *, u_int, u_int, struct uni_msg *) [static]

Definition at line 1103 of file ng_ccatm.c.

References uni_arg::cookie, chook::hook, NG_SEND_DATA_ONLY, uni_arg::sig, uni_msg_destroy(), and uni_msg_pack_mbuf().

Referenced by ng_ccatm_log().

Here is the call graph for this function:



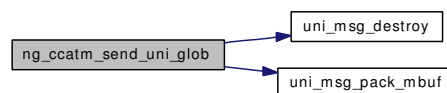
7.5.1.21 `static void ng_ccatm_send_uni_glob (struct ccport *, void *, u_int, u_int, struct uni_msg *)` [static]

Definition at line 1128 of file `ng_ccatm.c`.

References `uni_arg::cookie`, `cchook::hook`, `NG_SEND_DATA_ONLY`, `uni_arg::sig`, `uni_msg_destroy()`, and `uni_msg_pack_mbuf()`.

Referenced by `ng_ccatm_log()`.

Here is the call graph for this function:



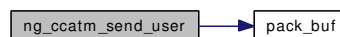
7.5.1.22 `static void ng_ccatm_send_user (struct ccuser *, void *, u_int, void *, size_t)` [static]

Definition at line 1006 of file `ng_ccatm.c`.

References `cchook::hook`, `NG_SEND_DATA_ONLY`, `ccatm_op::op`, and `pack_buf()`.

Referenced by `ng_ccatm_log()`.

Here is the call graph for this function:



7.5.1.23 `static int ng_ccatm_shutdown (node_p node)` [static]

Definition at line 374 of file `ng_ccatm.c`.

References `ccnode::data`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `ccnode::node`.

7.5.1.24 `static struct mbuf* pack_buf (void * h, size_t hlen, void * t, size_t tlen)` [static]

Definition at line 956 of file `ng_ccatm.c`.

Referenced by `ng_ccatm_respond_user()`, and `ng_ccatm_send_user()`.

7.5.1.25 `static int send_dump (struct ccdata * data, void * uarg, const char * buf)` [static]

Definition at line 433 of file `ng_ccatm.c`.

References `ccnode::dump`, `ccnode::dump_first`, and `ccnode::dump_last`.

Referenced by `ng_ccatm_dump()`.

7.5.2 Variable Documentation

7.5.2.1 struct [ng_parse_type ng_ccatm_addr_array_type](#) [static]

Initial value:

```
{
    &ng_parse_fixedarray_type,
    &ng_ccatm_addr_array_type_info
}
```

Definition at line 100 of file `ng_ccatm.c`.

7.5.2.2 struct [ng_parse_fixedarray_info ng_ccatm_addr_array_type_info](#) [static]

Initial value:

```
NGM_CCATM_ADDR_ARRAY_INFO
```

Definition at line 98 of file `ng_ccatm.c`.

7.5.2.3 struct [ng_parse_type ng_ccatm_addr_req_array_type](#) [static]

Initial value:

```
{
    &ng_parse_array_type,
    &ng_ccatm_addr_req_array_type_info
}
```

Definition at line 134 of file `ng_ccatm.c`.

7.5.2.4 struct [ng_parse_array_info ng_ccatm_addr_req_array_type_info](#) [static]

Initial value:

```
NGM_CCATM_ADDR_REQ_ARRAY_INFO
```

Definition at line 132 of file `ng_ccatm.c`.

7.5.2.5 struct [ng_parse_type ng_ccatm_addr_req_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_ccatm_addr_req_type_info
}
```

Definition at line 116 of file `ng_ccatm.c`.

7.5.2.6 struct ng_parse_struct_field ng_ccatm_addr_req_type_info[] [static]**Initial value:**

```
NGM_CCATM_ADDR_REQ_INFO
```

Definition at line 114 of file ng_ccatm.c.

7.5.2.7 struct ng_parse_type ng_ccatm_atm_port_type [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_ccatm_atm_port_type_info
}
```

Definition at line 84 of file ng_ccatm.c.

7.5.2.8 struct ng_parse_struct_field ng_ccatm_atm_port_type_info[] [static]**Initial value:**

```
NGM_CCATM_ATM_PORT_INFO
```

Definition at line 82 of file ng_ccatm.c.

7.5.2.9 struct ng_cmdlist ng_ccatm_cmdlist[] [static]

Definition at line 176 of file ng_ccatm.c.

7.5.2.10 ng_constructor_t ng_ccatm_constructor [static]

Definition at line 274 of file ng_ccatm.c.

7.5.2.11 ng_disconnect_t ng_ccatm_disconnect [static]

Definition at line 279 of file ng_ccatm.c.

7.5.2.12 struct ng_parse_type ng_ccatm_esi_type [static]**Initial value:**

```
{
    &ng_parse_fixedarray_type,
    &ng_ccatm_esi_type_info
}
```

Definition at line 76 of file ng_ccatm.c.

7.5.2.13 `struct ng_parse_fixedarray_info ng_ccatm_esi_type_info` [static]**Initial value:**

```
NGM_CCATM_ESI_INFO
```

Definition at line 74 of file ng_ccatm.c.

7.5.2.14 `struct ng_parse_type ng_ccatm_get_addresses_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_ccatm_get_addresses_type_info
}
```

Definition at line 142 of file ng_ccatm.c.

7.5.2.15 `struct ng_parse_struct_field ng_ccatm_get_addresses_type_info[]` [static]**Initial value:**

```
NGM_CCATM_GET_ADDRESSES_INFO
```

Definition at line 140 of file ng_ccatm.c.

7.5.2.16 `ng_newhook_t ng_ccatm_newhook` [static]

Definition at line 277 of file ng_ccatm.c.

7.5.2.17 `struct ng_parse_type ng_ccatm_port_array_type` [static]**Initial value:**

```
{
    &ng_parse_array_type,
    &ng_ccatm_port_array_type_info
}
```

Definition at line 160 of file ng_ccatm.c.

7.5.2.18 `struct ng_parse_array_info ng_ccatm_port_array_type_info` [static]**Initial value:**

```
NGM_CCATM_PORT_ARRAY_INFO
```

Definition at line 158 of file ng_ccatm.c.

7.5.2.19 `struct ng_parse_type ng_ccatm_port_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_ccatm_port_type_info
}
```

Definition at line 92 of file ng_ccatm.c.

7.5.2.20 `struct ng_parse_struct_field ng_ccatm_port_type_info[]` [static]**Initial value:**

```
NGM_CCATM_PORT_INFO
```

Definition at line 90 of file ng_ccatm.c.

7.5.2.21 `struct ng_parse_type ng_ccatm_portlist_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_ccatm_portlist_type_info
}
```

Definition at line 168 of file ng_ccatm.c.

7.5.2.22 `struct ng_parse_struct_field ng_ccatm_portlist_type_info[]` [static]**Initial value:**

```
NGM_CCATM_PORTLIST_INFO
```

Definition at line 166 of file ng_ccatm.c.

7.5.2.23 `ng_rcvdata_t ng_ccatm_rcvdata` [static]

Definition at line 278 of file ng_ccatm.c.

7.5.2.24 `ng_rcvdata_t ng_ccatm_rcvdump` [static]

Definition at line 297 of file ng_ccatm.c.

Referenced by ng_ccatm_newhook().

7.5.2.25 `ng_rcvdata_t ng_ccatm_rcvmanage` [static]

Definition at line 298 of file ng_ccatm.c.

Referenced by ng_ccatm_newhook().

7.5.2.26 [ng_rcvmsg_t ng_ccatm_rcvmsg](#) [static]

Definition at line 275 of file ng_ccatm.c.

7.5.2.27 [ng_rcvdata_t ng_ccatm_rcvuni](#) [static]

Definition at line 296 of file ng_ccatm.c.

Referenced by ng_ccatm_newhook().

7.5.2.28 [ng_shutdown_t ng_ccatm_shutdown](#) [static]

Definition at line 276 of file ng_ccatm.c.

7.5.2.29 [struct ng_type ng_ccatm_typestruct](#) [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_CCATM_NODE_TYPE,
    .mod_event =   ng_ccatm_mod_event,
    .constructor = ng_ccatm_constructor,
    .rcvmsg =      ng_ccatm_rcvmsg,
    .shutdown =    ng_ccatm_shutdown,
    .newhook =     ng_ccatm_newhook,
    .rcvdata =     ng_ccatm_rcvdata,
    .disconnect =  ng_ccatm_disconnect,
    .cmdlist =     ng_ccatm_cmdlist,
}
```

Definition at line 282 of file ng_ccatm.c.

7.5.2.30 [struct ng_parse_type ng_ccatm_uni_addr_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_ccatm_uni_addr_type_info
}
```

Definition at line 108 of file ng_ccatm.c.

7.5.2.31 [struct ng_parse_struct_field ng_ccatm_uni_addr_type_info\[\]](#) [static]**Initial value:**

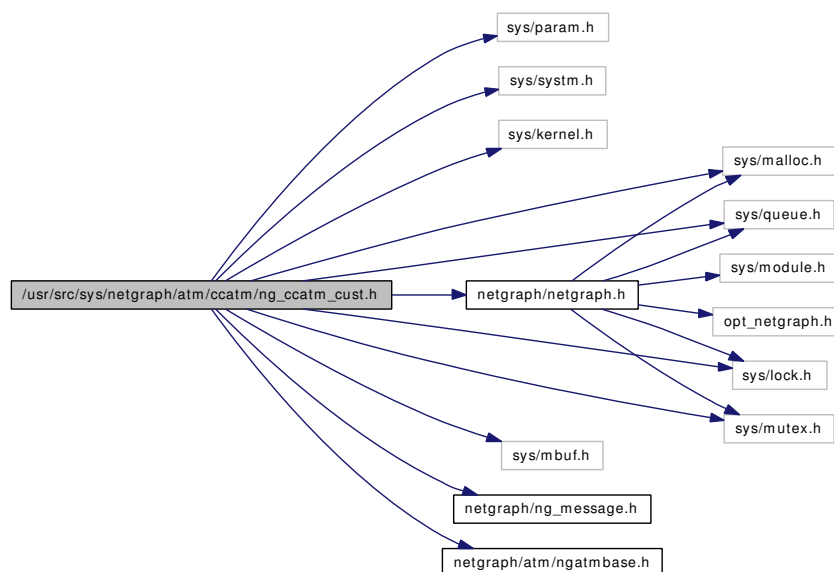
```
NGM_CCATM_UNI_ADDR_INFO
```

Definition at line 106 of file ng_ccatm.c.

7.6 /usr/src/sys/netgraph/atm/ccatm/ng_ccatm_cust.h File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/queue.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/atm/ngatmbase.h>
```

Include dependency graph for ng_ccatm_cust.h:



Defines

- #define **CCASSERT**(E, M) KASSERT(E, M)
- #define **CCMALLOC**(S) (malloc((S), M_NG_CCATM, M_NOWAIT))
- #define **CCZALLOC**(S) (malloc((S), M_NG_CCATM, M_NOWAIT | M_ZERO))
- #define **CCFREE**(P) do { free((P), M_NG_CCATM); } while (0)
- #define **CCGETERRNO**() (ENOMEM)

Functions

- [MALLOC_DECLARE](#) (M_NG_CCATM)

7.6.1 Define Documentation

7.6.1.1 #define CCASSERT(E, M) KASSERT(E, M)

Definition at line 46 of file ng_ccatm_cust.h.

7.6.1.2 #define CCFREE(P) do { free((P), M_NG_CCATM); } while (0)

Definition at line 52 of file ng_ccatm_cust.h.

7.6.1.3 #define CCGETERRNO() (ENOMEM)

Definition at line 54 of file ng_ccatm_cust.h.

7.6.1.4 #define CCMALLOC(S) (malloc((S), M_NG_CCATM, M_NOWAIT))

Definition at line 50 of file ng_ccatm_cust.h.

7.6.1.5 #define CCZALLOC(S) (malloc((S), M_NG_CCATM, M_NOWAIT | M_ZERO))

Definition at line 51 of file ng_ccatm_cust.h.

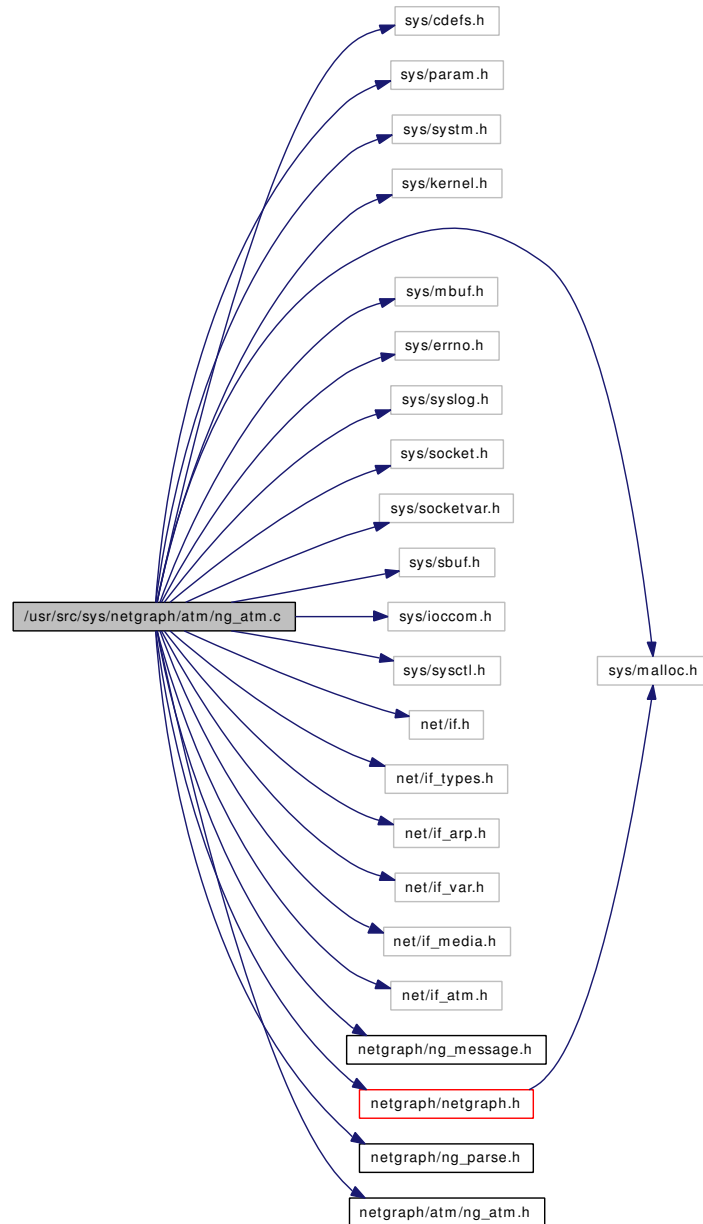
7.6.2 Function Documentation

7.6.2.1 MALLOC_DECLARE (M_NG_CCATM)

7.7 /usr/src/sys/netgraph/atm/ng_atm.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sbuf.h>
#include <sys/ioccom.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/if_arp.h>
#include <net/if_var.h>
#include <net/if_media.h>
#include <net/if_atm.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/atm/ng_atm.h>
```

Include dependency graph for ng_atm.c:



Data Structures

- struct [ngvcc](#)
- struct [priv](#)

Defines

- #define [VCC_OPEN](#) 0x0001
- #define [IFP2NG](#)(IFP) (([node_p](#))((struct ifatm *) (IFP) → if_softc) → ngpriv)
- #define [IFP2NG_SET](#)(IFP, val) (((struct ifatm *) (IFP) → if_softc) → ngpriv = (val))
- #define [IFFLAGS](#)

Functions

- `__FBSDDID` ("FreeBSD: src/sys/netgraph/atm/ng_atm.c,v 1.15 2005/08/10 06:25:40 obrien Exp \$")
- `SYSCTL_NODE` (`_net_graph`, `OID_AUTO`, `atm`, `CTLFLAG_RW`, 0, "atm related stuff")
- static int `ng_atm_vccarray_getlen` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf)
- static int `ng_atm_mod_event` (module_t, int, void *)
- `NETGRAPH_INIT` (atm, &`ng_atm_tpestruct`)
- static void `ng_atm_input` (struct ifnet *ifp, struct mbuf **mp, struct atm_pseudohdr *ah, void *rxhand)
- static int `ng_atm_output` (struct ifnet *ifp, struct mbuf **mp)
- static void `ng_atm_input_orphans` (struct ifnet *ifp, struct mbuf *m, struct atm_pseudohdr *ah, void *rxhand)
- static int `ng_atm_rcvdata` (hook_p hook, item_p item)
- static int `ng_atm_rcvdrop` (hook_p hook, item_p item)
- static void `ng_atm_event_func` (node_p node, hook_p hook, void *arg, int event)
- static void `ng_atm_event` (struct ifnet *ifp, uint32_t event, void *arg)
- static int `ng_atm_cpcs_init` (node_p node, const struct `ngm_atm_cpcs_init` *arg)
- static int `cpcs_term` (const struct `priv` *priv, u_int vpi, u_int vci)
- static int `ng_atm_cpcs_term` (node_p node, const struct `ngm_atm_cpcs_term` *arg)
- static int `text_status` (node_p node, char *arg, u_int len)
- static int `ng_atm_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_atm_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_atm_connect` (hook_p hook)
- static int `ng_atm_disconnect` (hook_p hook)
- static void `ng_atm_attach` (struct ifnet *ifp)
- static void `ng_atm_detach` (struct ifnet *ifp)
- static int `ng_atm_shutdown` (node_p node)
- static int `ng_atm_constructor` (node_p nodep)

Variables

- void(*) `ng_atm_attach_p` (struct ifnet *)
- void(*) `ng_atm_detach_p` (struct ifnet *)
- int(*) `ng_atm_output_p` (struct ifnet *, struct mbuf **)
- void(*) `ng_atm_input_p` (struct ifnet *, struct mbuf **, struct atm_pseudohdr *, void *)
- void(*) `ng_atm_input_orphan_p` (struct ifnet *, struct mbuf *, struct atm_pseudohdr *, void *)
- void(*) `ng_atm_event_p` (struct ifnet *, uint32_t, void *)
- static struct `ng_parse_struct_field` `ng_atm_if_change_info` []
- static struct `ng_parse_type` `ng_atm_if_change_type`
- static struct `ng_parse_struct_field` `ng_atm_vcc_change_info` []
- static struct `ng_parse_type` `ng_atm_vcc_change_type`
- static struct `ng_parse_struct_field` `ng_atm_acr_change_info` []
- static struct `ng_parse_type` `ng_atm_acr_change_type`
- static struct `ng_parse_struct_field` `ng_atm_config_type_info` []
- static struct `ng_parse_type` `ng_atm_config_type`
- static struct `ng_parse_struct_field` `ng_atm_tparam_type_info` []
- static struct `ng_parse_type` `ng_atm_tparam_type`
- static struct `ng_parse_struct_field` `ng_atm_vcc_type_info` []
- static struct `ng_parse_type` `ng_atm_vcc_type`

- static struct `ng_parse_array_info ng_atm_vccarray_info`
- static struct `ng_parse_type ng_atm_vccarray_type`
- static struct `ng_parse_struct_field ng_atm_vcctable_type_info []`
- static struct `ng_parse_type ng_atm_vcctable_type`
- static struct `ng_parse_struct_field ng_atm_cpcs_init_type_info []`
- static struct `ng_parse_type ng_atm_cpcs_init_type`
- static struct `ng_parse_struct_field ng_atm_cpcs_term_type_info []`
- static struct `ng_parse_type ng_atm_cpcs_term_type`
- static struct `ng_parse_struct_field ng_atm_stats_type_info []`
- static struct `ng_parse_type ng_atm_stats_type`
- static struct `ng_cmdlist ng_atm_cmdlist []`
- static `ng_constructor_t ng_atm_constructor`
- static `ng_shutdown_t ng_atm_shutdown`
- static `ng_rcvmsg_t ng_atm_rcvmsg`
- static `ng_newhook_t ng_atm_newhook`
- static `ng_connect_t ng_atm_connect`
- static `ng_disconnect_t ng_atm_disconnect`
- static `ng_rcvdata_t ng_atm_rcvdata`
- static `ng_rcvdata_t ng_atm_rcvdrop`
- static struct `ng_type ng_atm_typestruct`
- struct {
 - u_int `media`
 - const char * `name`
- } `atmmmedia []`

7.7.1 Define Documentation

7.7.1.1 #define IFFLAGS

Value:

```
"\020\001UP\002BROADCAST\003DEBUG\004LOOPBACK" \
    "\005POINTOPOINT\006SMART\007RUNNING\010NOARP" \
    "\011PROMISC\012ALLMULTI\013ACTIVE\014SIMPLEX" \
    "\015LINK0\016LINK1\017LINK2\020MULTICAST"
```

Definition at line 352 of file `ng_atm.c`.

7.7.1.2 #define IFP2NG(IFP) ((**node_p**)((struct ifatm *) (IFP) → if_softc) → ngpriv)

Definition at line 349 of file `ng_atm.c`.

Referenced by `ng_atm_attach()`, `ng_atm_detach()`, `ng_atm_event()`, `ng_atm_input()`, `ng_atm_input_orphans()`, `ng_atm_output()`, `ng_ether_attach()`, `ng_ether_detach()`, `ng_ether_input()`, `ng_ether_input_orphan()`, `ng_ether_link_state()`, `ng_ether_output()`, `ng_fec_addport()`, `ng_fec_delport()`, `ng_fec_input()`, `ng_gif_attach()`, `ng_gif_detach()`, `ng_gif_input()`, and `ng_gif_input_orphan()`.

7.7.1.3 #define IFP2NG_SET(IFP, val) (((struct ifatm *) (IFP) → if_softc) → ngpriv = (val))

Definition at line 350 of file ng_atm.c.

Referenced by ng_atm_attach(), ng_atm_detach(), ng_atm_shutdown(), ng_gif_attach(), and ng_gif_detach().

7.7.1.4 #define VCC_OPEN 0x0001

Definition at line 97 of file ng_atm.c.

Referenced by ng_atm_cpcs_init(), ng_atm_cpcs_term(), and ng_atm_disconnect().

7.7.2 Function Documentation**7.7.2.1 __FBSDID ("FreeBSD: src/sys/netgraph/atm/ng_atm.c, v 1.15 2005/08/10 06:25:40 obrien Exp \$")****7.7.2.2 static int cpcs_term (const struct priv *priv, u_int vpi, u_int vci) [static]**

Definition at line 803 of file ng_atm.c.

References priv::ifp.

Referenced by ng_atm_cpcs_term(), and ng_atm_disconnect().

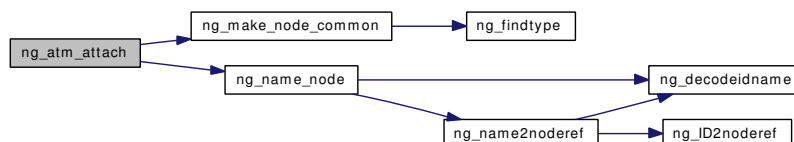
7.7.2.3 NETGRAPH_INIT (atm, & ng_atm_tpestruct)**7.7.2.4 static void ng_atm_attach (struct ifnet *ifp) [static]**

Definition at line 1260 of file ng_atm.c.

References IFP2NG, IFP2NG_SET, ng_atm_tpestruct, ng_make_node_common(), ng_name_node(), NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

Referenced by ng_atm_mod_event().

Here is the call graph for this function:

**7.7.2.5 static int ng_atm_connect (hook_p hook) [static]**

Definition at line 1202 of file ng_atm.c.

References ngvcc::hook, NG_HOOK_FORCE_QUEUE, NG_HOOK_PEER, and NG_HOOK_PRIVATE.

7.7.2.6 static int ng_atm_constructor (node_p nodep) [static]

Definition at line 1355 of file ng_atm.c.

7.7.2.7 static int ng_atm_cpcs_init (node_p node, const struct ngm_atm_cpcs_init * arg) [static]

Definition at line 657 of file ng_atm.c.

References ngm_atm_cpcs_init::aal, ngm_atm_cpcs_init::adtf, ngm_atm_cpcs_init::cdf, ngm_atm_cpcs_init::flags, ngvcc::flags, ngvcc::hook, ngm_atm_cpcs_init::icr, priv::ifp, ngm_atm_cpcs_init::mbs, ngm_atm_cpcs_init::mcr, ngm_atm_cpcs_init::name, NG_HOOK_NAME, NG_NODE_PRIVATE, ngm_atm_cpcs_init::nrm, ngm_atm_cpcs_init::pcr, ngm_atm_cpcs_init::rdf, ngm_atm_cpcs_init::rif, ngm_atm_cpcs_init::rmtu, ngm_atm_cpcs_init::scr, ngm_atm_cpcs_init::tbe, ngm_atm_cpcs_init::tmtu, ngm_atm_cpcs_init::traffic, ngm_atm_cpcs_init::trm, VCC_OPEN, ngvcc::vci, ngm_atm_cpcs_init::vci, ngvcc::vpi, and ngm_atm_cpcs_init::vpi.

Referenced by ng_atm_rcvmsg().

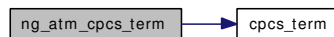
7.7.2.8 static int ng_atm_cpcs_term (node_p node, const struct ngm_atm_cpcs_term * arg) [static]

Definition at line 822 of file ng_atm.c.

References cpcs_term(), ngvcc::flags, ngvcc::hook, ngm_atm_cpcs_term::name, NG_HOOK_NAME, NG_NODE_PRIVATE, VCC_OPEN, ngvcc::vci, and ngvcc::vpi.

Referenced by ng_atm_rcvmsg().

Here is the call graph for this function:



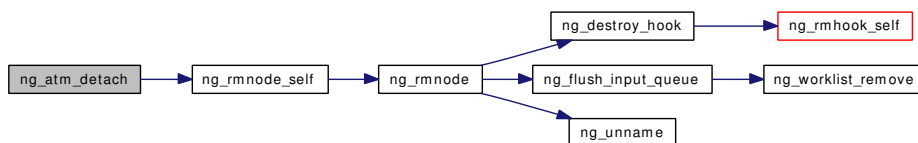
7.7.2.9 static void ng_atm_detach (struct ifnet * ifp) [static]

Definition at line 1295 of file ng_atm.c.

References priv::ifp, IFP2NG, IFP2NG_SET, NG_NODE_PRIVATE, NG_NODE_REALLY_DIE, and ng_rmnode_self().

Referenced by ng_atm_mod_event().

Here is the call graph for this function:

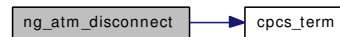


7.7.2.10 static int ng_atm_disconnect (hook_p hook) [static]

Definition at line 1214 of file ng_atm.c.

References `cpcs_term()`, `ngvcc::flags`, `ngvcc::hook`, `priv::ifp`, `priv::input`, `NG_HOOK_NAME`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_HOOK_SET_PRIVATE`, `NG_NODE_PRIVATE`, `priv::orphans`, `priv::output`, `VCC_OPEN`, `ngvcc::vci`, and `ngvcc::vpi`.

Here is the call graph for this function:

**7.7.2.11 static void ng_atm_event (struct ifnet * ifp, uint32_t event, void * arg) [static]**

Definition at line 640 of file ng_atm.c.

References `IFP2NG`, `ng_atm_event_func()`, `ng_send_fn`, and `ngm_atm_acr_change::node`.

Referenced by `ng_atm_mod_event()`.

Here is the call graph for this function:

**7.7.2.12 static void ng_atm_event_func (node_p node, hook_p hook, void * arg, int event) [static]**

Definition at line 533 of file ng_atm.c.

References `ngm_atm_acr_change::acr`, `ngm_atm_if_change::carrier`, `ng_mesg::data`, `ngvcc::hook`, `priv::manage`, `NG_MKMESSAGE`, `NG_NODE_ID`, `NG_NODE_PRIVATE`, `NG_SEND_MSG_HOOK`, `NGM_ATM_ACR_CHANGE`, `NGM_ATM_COOKIE`, `NGM_ATM_IF_CHANGE`, `NGM_ATM_VCC_CHANGE`, `NGM_FLOW_COOKIE`, `NGM_HIGH_WATER_PASSED`, `NGM_LOW_WATER_PASSED`, `ngm_atm_acr_change::node`, `ngm_atm_if_change::node`, `ngm_atm_vcc_change::node`, `ngm_atm_if_change::running`, `ngm_atm_vcc_change::state`, `ngm_atm_vcc_change::vci`, `ngvcc::vci`, `ngm_atm_vcc_change::vpi`, and `ngvcc::vpi`.

Referenced by `ng_atm_event()`.

7.7.2.13 static void ng_atm_input (struct ifnet * ifp, struct mbuf ** mp, struct atm_pseudohdr * ah, void * rxhand) [static]

Definition at line 370 of file ng_atm.c.

References `ngvcc::hook`, `IFP2NG`, `priv::in_errors`, `priv::in_packets`, `priv::input`, `NG_NODE_PRIVATE`, and `NG_SEND_DATA_ONLY`.

Referenced by `ng_atm_mod_event()`.

7.7.2.14 `static void ng_atm_input_orphans (struct ifnet * ifp, struct mbuf * m, struct atm_pseudohdr * ah, void * rxhand) [static]`

Definition at line 444 of file ng_atm.c.

References IFP2NG, priv::in_errors, priv::in_packets, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, and priv::orphans.

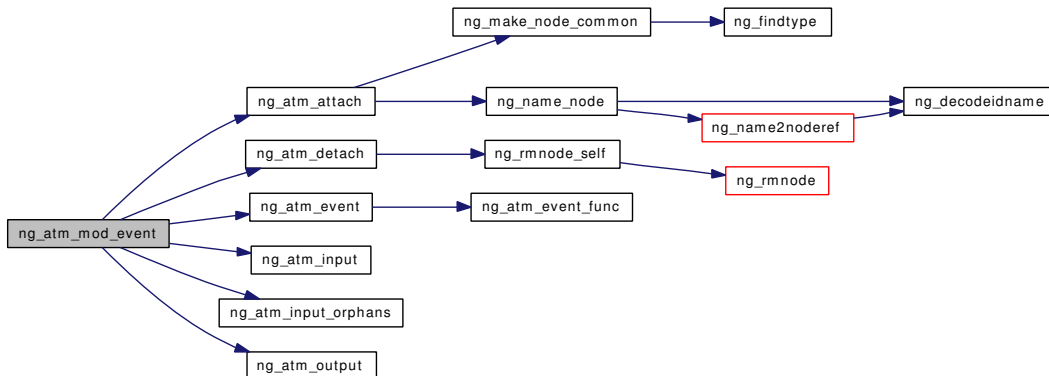
Referenced by ng_atm_mod_event().

7.7.2.15 `static int ng_atm_mod_event (module_t, int, void *) [static]`

Definition at line 1379 of file ng_atm.c.

References ng_atm_attach(), ng_atm_attach_p, ng_atm_detach(), ng_atm_detach_p, ng_atm_event(), ng_atm_event_p, ng_atm_input(), ng_atm_input_orphan_p, ng_atm_input_orphans(), ng_atm_input_p, ng_atm_output(), and ng_atm_output_p.

Here is the call graph for this function:



7.7.2.16 `static int ng_atm_newhook (node_p node, hook_p hook, const char * name) [static]`

Definition at line 1159 of file ng_atm.c.

References ngvcc::hook, priv::input, priv::manage, ng_atm_rcvdrop, NG_HOOK_SET_PRIVATE, NG_HOOK_SET_RCVDATA, NG_NODE_PRIVATE, priv::orphans, and priv::output.

7.7.2.17 `static int ng_atm_output (struct ifnet * ifp, struct mbuf ** mp) [static]`

Definition at line 423 of file ng_atm.c.

References IFP2NG, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, and priv::output.

Referenced by ng_atm_mod_event().

7.7.2.18 `static int ng_atm_rcvdata (hook_p hook, item_p item) [static]`

Definition at line 483 of file ng_atm.c.

References `ngvcc::hook`, `priv::ifp`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `priv::out_errors`, `priv::out_packets`, `ngvcc::vci`, and `ngvcc::vpi`.

7.7.2.19 `static int ng_atm_rcvdrop (hook_p hook, item_p item)` [static]

Definition at line 521 of file `ng_atm.c`.

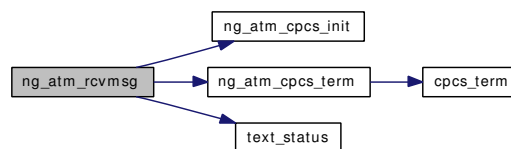
References `NG_FREE_ITEM`.

7.7.2.20 `static int ng_atm_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 903 of file `ng_atm.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `ngvcc::hook`, `priv::ifp`, `priv::in_errors`, `priv::in_packets`, `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `NG_HOOK_NAME`, `NG_HOOKSIZ`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_TEXTRESPONSE`, `NGI_GET_MSG`, `NGM_ATM_COOKIE`, `NGM_ATM_CPCS_INIT`, `NGM_ATM_CPCS_TERM`, `NGM_ATM_GET_CONFIG`, `NGM_ATM_GET_IFNAME`, `NGM_ATM_GET_STATS`, `NGM_ATM_GET_VCC`, `NGM_ATM_GET_VCCID`, `NGM_ATM_GET_VCCS`, `NGM_GENERIC_COOKIE`, `NGM_TEXT_STATUS`, `priv::out_errors`, `priv::out_packets`, `text_status()`, `ng_mesg::ng_msghdr::typecookie`, `ngvcc::vci`, and `ngvcc::vpi`.

Here is the call graph for this function:



7.7.2.21 `static int ng_atm_shutdown (node_p node)` [static]

Definition at line 1316 of file `ng_atm.c`.

References `priv::ifp`, `IFP2NG_SET`, `ng_node::nd_flags`, `NG_NODE_PRIVATE`, `NG_NODE_REVIVE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `NGF_REALLY_DIE`.

7.7.2.22 `static int ng_atm_vcarray_getlen (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 175 of file `ng_atm.c`.

7.7.2.23 `SYCTL_NODE (_net_graph, OID_AUTO, atm, CTLFLAG_RW, 0, "atm related stuff")`

7.7.2.24 `static int text_status (node_p node, char * arg, u_int len)` [static]

Definition at line 854 of file `ng_atm.c`.

References `atmmmedia`, `priv::ifp`, `name`, and `NG_NODE_PRIVATE`.

Referenced by `ng_atm_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, and `ng_uni_rcvmsg()`.

7.7.3 Variable Documentation

7.7.3.1 struct { ... } **atmmmedia**[] [static]

Referenced by `text_status()`.

7.7.3.2 **u_int media**

Definition at line 344 of file `ng_atm.c`.

7.7.3.3 **const char* name**

Definition at line 345 of file `ng_atm.c`.

Referenced by `ng_bridge_nodename()`, `ng_h4_ioctl()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_ksocket_newhook()`, `ng_ksocket_parse()`, `ng_unparse_composite()`, `ngt_open()`, and `text_status()`.

7.7.3.4 struct **ng_parse_struct_field ng_atm_acr_change_info**[] [static]

Initial value:

```
NGM_ATM_ACR_CHANGE_INFO
```

Definition at line 139 of file `ng_atm.c`.

7.7.3.5 struct **ng_parse_type ng_atm_acr_change_type** [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_acr_change_info
}
```

Definition at line 141 of file `ng_atm.c`.

7.7.3.6 **void(*) ng_atm_attach_p(struct ifnet *)**

Referenced by `ng_atm_mod_event()`.

7.7.3.7 struct **ng_cmdlist ng_atm_cmdlist**[] [static]

Definition at line 234 of file `ng_atm.c`.

7.7.3.8 struct **ng_parse_type ng_atm_config_type** [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_config_type_info
}
```

Definition at line 152 of file ng_atm.c.

7.7.3.9 struct [ng_parse_struct_field ng_atm_config_type_info](#)[] [static]

Initial value:

```
NGM_ATM_CONFIG_INFO
```

Definition at line 149 of file ng_atm.c.

7.7.3.10 [ng_connect_t ng_atm_connect](#) [static]

Definition at line 323 of file ng_atm.c.

7.7.3.11 [ng_constructor_t ng_atm_constructor](#) [static]

Definition at line 319 of file ng_atm.c.

7.7.3.12 struct [ng_parse_type ng_atm_cpcs_init_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_cpcs_init_type_info
}
```

Definition at line 207 of file ng_atm.c.

7.7.3.13 struct [ng_parse_struct_field ng_atm_cpcs_init_type_info](#)[] [static]

Initial value:

```
NGM_ATM_CPCS_INIT_INFO
```

Definition at line 204 of file ng_atm.c.

7.7.3.14 struct [ng_parse_type ng_atm_cpcs_term_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_cpcs_term_type_info
}
```

Definition at line 218 of file ng_atm.c.

7.7.3.15 `struct ng_parse_struct_field ng_atm_cpcs_term_type_info[]` [static]

Initial value:

```
NGM_ATM_CPCS_TERM_INFO
```

Definition at line 215 of file ng_atm.c.

7.7.3.16 `void(*) ng_atm_detach_p(struct ifnet *)`

Referenced by ng_atm_mod_event().

7.7.3.17 `ng_disconnect_t ng_atm_disconnect` [static]

Definition at line 324 of file ng_atm.c.

7.7.3.18 `void(*) ng_atm_event_p(struct ifnet *, uint32_t, void *)`

Referenced by ng_atm_mod_event().

7.7.3.19 `struct ng_parse_struct_field ng_atm_if_change_info[]` [static]

Initial value:

```
NGM_ATM_IF_CHANGE_INFO
```

Definition at line 119 of file ng_atm.c.

7.7.3.20 `struct ng_parse_type ng_atm_if_change_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_if_change_info
}
```

Definition at line 121 of file ng_atm.c.

7.7.3.21 `void(*) ng_atm_input_orphan_p(struct ifnet *, struct mbuf *, struct atm_pseudohdr *, void *)`

Referenced by ng_atm_mod_event().

7.7.3.22 `void(*) ng_atm_input_p(struct ifnet *, struct mbuf **, struct atm_pseudohdr *, void *)`

Referenced by ng_atm_mod_event().

7.7.3.23 `ng_newhook_t ng_atm_newhook` [static]

Definition at line 322 of file ng_atm.c.

7.7.3.24 `int(*) ng_atm_output_p(struct ifnet *, struct mbuf **)`

Referenced by ng_atm_mod_event().

7.7.3.25 `ng_rcvdata_t ng_atm_rcvdata` [static]

Definition at line 325 of file ng_atm.c.

7.7.3.26 `ng_rcvdata_t ng_atm_rcvdrop` [static]

Definition at line 326 of file ng_atm.c.

Referenced by ng_atm_newhook().

7.7.3.27 `ng_rcvmsg_t ng_atm_rcvmsg` [static]

Definition at line 321 of file ng_atm.c.

7.7.3.28 `ng_shutdown_t ng_atm_shutdown` [static]

Definition at line 320 of file ng_atm.c.

7.7.3.29 `struct ng_parse_type ng_atm_stats_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_atm_stats_type_info
}
```

Definition at line 229 of file ng_atm.c.

7.7.3.30 `struct ng_parse_struct_field ng_atm_stats_type_info[]` [static]**Initial value:**

```
NGM_ATM_STATS_INFO
```

Definition at line 226 of file ng_atm.c.

7.7.3.31 struct [ng_parse_type ng_atm_tparam_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_atm_tparam_type_info
}
```

Definition at line 162 of file `ng_atm.c`.

7.7.3.32 struct [ng_parse_struct_field ng_atm_tparam_type_info\[\]](#) [static]**Initial value:**

```
NGM_ATM_TPARAM_INFO
```

Definition at line 160 of file `ng_atm.c`.

7.7.3.33 struct [ng_type ng_atm_tpestruct](#) [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_ATM_NODE_TYPE,
    .mod_event =   ng_atm_mod_event,
    .constructor = ng_atm_constructor,
    .rcvmsg =      ng_atm_rcvmsg,
    .shutdown =    ng_atm_shutdown,
    .newhook =     ng_atm_newhook,
    .connect =     ng_atm_connect,
    .rcvdata =     ng_atm_rcvdata,
    .disconnect =  ng_atm_disconnect,
    .cmdlist =     ng_atm_cmdlist,
}
```

Definition at line 328 of file `ng_atm.c`.

Referenced by `ng_atm_attach()`.

7.7.3.34 struct [ng_parse_struct_field ng_atm_vcc_change_info\[\]](#) [static]**Initial value:**

```
NGM_ATM_VCC_CHANGE_INFO
```

Definition at line 129 of file `ng_atm.c`.

7.7.3.35 struct [ng_parse_type ng_atm_vcc_change_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_atm_vcc_change_info
}
```

Definition at line 131 of file ng_atm.c.

7.7.3.36 struct [ng_parse_type ng_atm_vcc_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_vcc_type_info
}
```

Definition at line 168 of file ng_atm.c.

7.7.3.37 struct [ng_parse_struct_field ng_atm_vcc_type_info](#)[] [static]

Initial value:

```
NGM_ATM_VCC_INFO
```

Definition at line 166 of file ng_atm.c.

7.7.3.38 struct [ng_parse_array_info ng_atm_vccarray_info](#) [static]

Initial value:

```
NGM_ATM_VCCARRAY_INFO
```

Definition at line 185 of file ng_atm.c.

7.7.3.39 struct [ng_parse_type ng_atm_vccarray_type](#) [static]

Initial value:

```
{
    &ng_parse_array_type,
    &ng_atm_vccarray_info
}
```

Definition at line 187 of file ng_atm.c.

7.7.3.40 struct [ng_parse_type ng_atm_vcctable_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_atm_vcctable_type_info
}
```

Definition at line 196 of file ng_atm.c.

7.7.3.41 `struct ng_parse_struct_field ng_atm_vcctable_type_info[]` `[static]`

Initial value:

`NGM_ATM_VCCTABLE_INFO`

Definition at line 193 of file `ng_atm.c`.

7.8 /usr/src/sys/netgraph/atm/ng_atm.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngm_atm_config](#)
- struct [ngm_atm_cpcs_init](#)
- struct [ngm_atm_cpcs_term](#)
- struct [ngm_atm_stats](#)
- struct [ngm_atm_if_change](#)
- struct [ngm_atm_vcc_change](#)
- struct [ngm_atm_acr_change](#)

Defines

- #define [NG_ATM_NODE_TYPE](#) "atm"
- #define [NGM_ATM_COOKIE](#) 960802260
- #define [NGM_ATM_CONFIG_INFO](#)
- #define [NGM_ATM_TPARAM_INFO](#)
- #define [NGM_ATM_VCC_INFO](#)
- #define [NGM_ATM_VCCARRAY_INFO](#)
- #define [NGM_ATM_VCCTABLE_INFO](#)
- #define [NGM_ATM_CPCS_INIT_INFO](#)
- #define [NGM_ATM_CPCS_TERM_INFO](#)
- #define [NGM_ATM_STATS_INFO](#)
- #define [NGM_ATM_IF_CHANGE_INFO](#)
- #define [NGM_ATM_VCC_CHANGE_INFO](#)
- #define [NGM_ATM_ACR_CHANGE_INFO](#)

Enumerations

- enum {
 [NGM_ATM_GET_IFNAME](#) = 1, [NGM_ATM_GET_CONFIG](#), [NGM_ATM_GET_VCCS](#), [NGM_ATM_CPCS_INIT](#),
 [NGM_ATM_CPCS_TERM](#), [NGM_ATM_GET_VCC](#), [NGM_ATM_GET_VCCID](#), [NGM_ATM_GET_STATS](#),
 [NGM_ATM_CARRIER_CHANGE](#) = 1000, [NGM_ATM_VCC_CHANGE](#), [NGM_ATM_ACR_CHANGE](#), [NGM_ATM_IF_CHANGE](#) }

7.8.1 Define Documentation

7.8.1.1 #define NG_ATM_NODE_TYPE "atm"

Definition at line 38 of file ng_atm.h.

7.8.1.2 #define NGM_ATM_ACR_CHANGE_INFO**Value:**

```

{
    { "node",      &ng_parse_hint32_type }, \
    { "vci",      &ng_parse_uint16_type }, \
    { "vpi",      &ng_parse_uint8_type }, \
    { "acr",      &ng_parse_uint32_type }, \
    { NULL } \
}

```

Definition at line 239 of file ng_atm.h.

7.8.1.3 #define NGM_ATM_CONFIG_INFO**Value:**

```

{
    { "pcr",      &ng_parse_uint32_type }, \
    { "vpi_bits", &ng_parse_uint32_type }, \
    { "vci_bits", &ng_parse_uint32_type }, \
    { "max_vpcs", &ng_parse_uint32_type }, \
    { "max_vccs", &ng_parse_uint32_type }, \
    { NULL } \
}

```

Definition at line 69 of file ng_atm.h.

7.8.1.4 #define NGM_ATM_COOKIE 960802260

Definition at line 39 of file ng_atm.h.

Referenced by ng_atm_event_func(), and ng_atm_rcvmsg().

7.8.1.5 #define NGM_ATM_CPCS_INIT_INFO**Value:**

```

{
    { "name",      &ng_parse_hookbuf_type }, \
    { "flags",    &ng_parse_hint32_type }, \
    { "vci",      &ng_parse_uint16_type }, \
    { "vpi",      &ng_parse_uint16_type }, \
    { "rmtu",     &ng_parse_uint16_type }, \
    { "tmtu",     &ng_parse_uint16_type }, \
    { "aal",      &ng_parse_uint8_type }, \
    { "traffic",  &ng_parse_uint8_type }, \
    { "pcr",      &ng_parse_uint32_type }, \
    { "scr",      &ng_parse_uint32_type }, \
    { "mbs",      &ng_parse_uint32_type }, \
    { "mcr",      &ng_parse_uint32_type }, \
    { "icr",      &ng_parse_uint32_type }, \
    { "tbe",      &ng_parse_uint32_type }, \
    { "nrm",      &ng_parse_uint8_type }, \
    { "trm",      &ng_parse_uint8_type }, \
    { "adtf",    &ng_parse_uint16_type }, \
    { "rif",      &ng_parse_uint8_type }, \
}

```

```

    { "rdf",      &ng_parse_uint8_type }, \
    { "cdf",      &ng_parse_uint8_type }, \
    { NULL } \
}

```

Definition at line 153 of file ng_atm.h.

7.8.1.6 #define NGM_ATM_CPCS_TERM_INFO

Value:

```

{
    { "name",      &ng_parse_hookbuf_type }, \
    { NULL } \
}

```

Definition at line 184 of file ng_atm.h.

7.8.1.7 #define NGM_ATM_IF_CHANGE_INFO

Value:

```

{
    { "node",      &ng_parse_hint32_type }, \
    { "carrier",   &ng_parse_uint8_type }, \
    { "running",   &ng_parse_uint8_type }, \
    { NULL } \
}

```

Definition at line 210 of file ng_atm.h.

7.8.1.8 #define NGM_ATM_STATS_INFO

Value:

```

{
    { "in_packets", &ng_parse_uint64_type }, \
    { "in_errors",  &ng_parse_uint64_type }, \
    { "out_packets", &ng_parse_uint64_type }, \
    { "out_errors", &ng_parse_uint64_type }, \
    { NULL } \
}

```

Definition at line 196 of file ng_atm.h.

7.8.1.9 #define NGM_ATM_TPARAM_INFO

Value:

```

{
    { "pcr",      &ng_parse_uint32_type }, \
    { "scr",      &ng_parse_uint32_type }, \
    { "mbs",      &ng_parse_uint32_type }, \
    { "mcr",      &ng_parse_uint32_type }, \
}

```

```

    { "icr",      &ng_parse_uint32_type }, \
    { "tbe",      &ng_parse_uint32_type }, \
    { "nrm",      &ng_parse_uint8_type }, \
    { "trm",      &ng_parse_uint8_type }, \
    { "adtf",     &ng_parse_uint16_type }, \
    { "rif",      &ng_parse_uint8_type }, \
    { "rdf",      &ng_parse_uint8_type }, \
    { "cdf",      &ng_parse_uint8_type }, \
    { NULL }
}

```

Definition at line 83 of file ng_atm.h.

7.8.1.10 #define NGM_ATM_VCC_CHANGE_INFO

Value:

```

{
    { "node",      &ng_parse_hint32_type }, \
    { "vci",      &ng_parse_uint16_type }, \
    { "vpi",      &ng_parse_uint8_type }, \
    { "state",    &ng_parse_uint8_type }, \
    { NULL }
}

```

Definition at line 224 of file ng_atm.h.

7.8.1.11 #define NGM_ATM_VCC_INFO

Value:

```

{
    { "flags",    &ng_parse_hint16_type }, \
    { "vpi",     &ng_parse_uint16_type }, \
    { "vci",     &ng_parse_uint16_type }, \
    { "rmtu",    &ng_parse_uint16_type }, \
    { "tmtu",    &ng_parse_uint16_type }, \
    { "aal",     &ng_parse_uint8_type }, \
    { "traffic", &ng_parse_uint8_type }, \
    { "tparam",  &ng_atm_tparam_type }, \
    { NULL }
}

```

Definition at line 100 of file ng_atm.h.

7.8.1.12 #define NGM_ATM_VCCARRAY_INFO

Value:

```

{
    &ng_atm_vcc_type, \
    ng_atm_vccarray_getlen, \
    NULL
}

```

Definition at line 113 of file ng_atm.h.

7.8.1.13 #define NGM_ATM_VCCTABLE_INFO

Value:

```
{
    { "count",    &ng_parse_uint32_type },
    { "vccs",    &ng_atm_vccarray_type },
    { NULL }
}
```

Definition at line 120 of file ng_atm.h.

7.8.2 Enumeration Type Documentation

7.8.2.1 anonymous enum

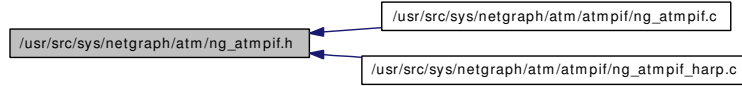
Enumerator:

NGM_ATM_GET_IFNAME
NGM_ATM_GET_CONFIG
NGM_ATM_GET_VCCS
NGM_ATM_CPCS_INIT
NGM_ATM_CPCS_TERM
NGM_ATM_GET_VCC
NGM_ATM_GET_VCCID
NGM_ATM_GET_STATS
NGM_ATM_CARRIER_CHANGE
NGM_ATM_VCC_CHANGE
NGM_ATM_ACR_CHANGE
NGM_ATM_IF_CHANGE

Definition at line 42 of file ng_atm.h.

7.9 /usr/src/sys/netgraph/atm/ng_atmpif.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_vatmpif_config](#)
- struct [hva_stats_ng](#)
- struct [hva_stats_atm](#)
- struct [hva_stats_aal5](#)
- struct [vatmpif_stats](#)
- struct [ng_atmpif_link_status](#)

Defines

- #define [NG_ATMPIF_NODE_TYPE](#) "atmpif"
- #define [NGM_ATMPIF_COOKIE](#) 967239456
- #define [VATMPIF_DEV_NAME](#) "hva"
- #define [VATMPIF_MAX_VCI](#) 65535
- #define [VATMPIF_MAX_VPI](#) 255
- #define [NG_ATMPIF_HOOK_LINK](#) "link"
- #define [NG_ATMPIF_CONFIG_TYPE_INFO](#)
- #define [VATMPIF_DEBUG_NONE](#) 0x00
- #define [VATMPIF_DEBUG_PACKET](#) 0x01
- #define [IS_VATMPIF_DEBUG_PACKET](#)(a)
- #define [HVA_STATS_NG_TYPE_INFO](#)
- #define [HVA_STATS_ATM_NG_TYPE_INFO](#)
- #define [HVA_STATS_AAL5_NG_TYPE_INFO](#)
- #define [NG_ATMPIF_STATS_TYPE_INFO](#)
- #define [NG_ATMPIF_LINK_STATUS_TYPE_INFO](#)

Typedefs

- typedef [hva_stats_ng](#) [Hva_Stats_ng](#)
- typedef [hva_stats_atm](#) [Hva_Stats_atm](#)
- typedef [hva_stats_aal5](#) [Hva_Stats_aal5](#)
- typedef [vatmpif_stats](#) [Vatmpif_stats](#)

Enumerations

- enum {
[NGM_ATMPIF_SET_CONFIG](#) = 1, [NGM_ATMPIF_GET_CONFIG](#), [NGM_ATMPIF_GET_-LINK_STATUS](#), [NGM_ATMPIF_GET_STATS](#),
[NGM_ATMPIF_CLR_STATS](#), [NGM_ATMPIF_GETCLR_STATS](#) }

7.9.1 Define Documentation

7.9.1.1 #define HVA_STATS_AAL5_NG_TYPE_INFO

Value:

```
{ "xmitAAL5Cells",          &ng_parse_uint64_type  },      \
  { "recvAAL5Cells",       &ng_parse_uint64_type  },      \
  { "AAL5ErrCRCCells",    &ng_parse_uint32_type  },      \
  { "AAL5DropsCells",     &ng_parse_uint32_type  },      \
  { "xmitAAL5PDU",        &ng_parse_uint64_type  },      \
  { "recvAAL5PDU",        &ng_parse_uint64_type  },      \
  { "AAL5CRCPDU",         &ng_parse_uint32_type  },      \
  { "AAL5ErrPDU",         &ng_parse_uint32_type  },      \
  { "AAL5DropsPDU",       &ng_parse_uint32_type  },      \
```

Definition at line 126 of file ng_atmpif.h.

7.9.1.2 #define HVA_STATS_ATM_NG_TYPE_INFO

Value:

```
{ "xmitATMCells",          &ng_parse_uint64_type  },      \
  { "recvATMCells",        &ng_parse_uint64_type  },      \
```

Definition at line 109 of file ng_atmpif.h.

7.9.1.3 #define HVA_STATS_NG_TYPE_INFO

Value:

```
{ "errSeqOrder",          &ng_parse_uint32_type  },      \
  { "errLostPDU",          &ng_parse_uint32_type  },      \
  { "recvBadPDU",          &ng_parse_uint32_type  },      \
  { "ErrATMVC",            &ng_parse_uint32_type  },      \
  { "ErrQfull",            &ng_parse_uint32_type  },      \
  { "xmitRawCell",         &ng_parse_uint32_type  },      \
  { "recvRawCell",         &ng_parse_uint32_type  },      \
  { "xmitPDU",             &ng_parse_uint64_type  },      \
  { "recvPDU",             &ng_parse_uint64_type  },      \
```

Definition at line 91 of file ng_atmpif.h.

7.9.1.4 #define IS_VATMPIF_DEBUG_PACKET(a)

Value:

```
( (a) \
                                     && ((a)->conf.debug & VATMPIF_DEBUG_PACKET) )
```

Definition at line 72 of file ng_atmpif.h.

Referenced by ng_atmpif_transmit(), vatmpif_harp_output(), and vatmpif_harp_recv_drain().

7.9.1.5 #define NG_ATMPIF_CONFIG_TYPE_INFO**Value:**

```
{
    { "debug",          \
      &ng_parse_uint8_type }, \
    { "pcr",           \
      &ng_parse_uint32_type }, \
    { "macaddr",      \
      &ng_mac_addr_type }, \
    { NULL }
}
```

Definition at line 59 of file ng_atmpif.h.

7.9.1.6 #define NG_ATMPIF_HOOK_LINK "link"

Definition at line 48 of file ng_atmpif.h.

Referenced by ng_atmpif_newhook().

7.9.1.7 #define NG_ATMPIF_LINK_STATUS_TYPE_INFO**Value:**

```
{
    { "InSeq",          \
      &ng_parse_uint32_type }, \
    { "OutSeq",        \
      &ng_parse_uint32_type }, \
    { "cur_pcr",       \
      &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 158 of file ng_atmpif.h.

7.9.1.8 #define NG_ATMPIF_NODE_TYPE "atmpif"

Definition at line 34 of file ng_atmpif.h.

7.9.1.9 #define NG_ATMPIF_STATS_TYPE_INFO**Value:**

```
{
    \
    HVA_STATS_NG_TYPE_INFO \
    HVA_STATS_ATM_NG_TYPE_INFO \
    HVA_STATS_AAL5_NG_TYPE_INFO \
    { NULL }
}
```

Definition at line 144 of file ng_atmpif.h.

7.9.1.10 #define NGM_ATMPIF_COOKIE 967239456

Definition at line 35 of file ng_atmpif.h.

Referenced by ng_atmpif_rcvmsg(), and vatmpif_harp_attach().

7.9.1.11 #define VATMPIF_DEBUG_NONE 0x00

Definition at line 69 of file ng_atmpif.h.

7.9.1.12 #define VATMPIF_DEBUG_PACKET 0x01

Definition at line 70 of file ng_atmpif.h.

7.9.1.13 #define VATMPIF_DEV_NAME "hva"

Definition at line 41 of file ng_atmpif.h.

Referenced by vatmpif_harp_attach().

7.9.1.14 #define VATMPIF_MAX_VCI 65535

Definition at line 44 of file ng_atmpif.h.

Referenced by vatmpif_harp_attach().

7.9.1.15 #define VATMPIF_MAX_VPI 255

Definition at line 45 of file ng_atmpif.h.

Referenced by vatmpif_harp_attach().

7.9.2 Typedef Documentation**7.9.2.1 typedef struct [hva_stats_aal5](#) Hva_Stats_aal5**

Definition at line 124 of file ng_atmpif.h.

7.9.2.2 typedef struct [hva_stats_atm](#) Hva_Stats_atm

Definition at line 107 of file ng_atmpif.h.

7.9.2.3 typedef struct [hva_stats_ng](#) Hva_Stats_ng

Definition at line 89 of file ng_atmpif.h.

7.9.2.4 typedef struct [vatmpif_stats](#) Vatmpif_stats

Definition at line 142 of file ng_atmpif.h.

7.9.3 Enumeration Type Documentation

7.9.3.1 anonymous enum

Enumerator:

NGM_ATMPIF_SET_CONFIG
NGM_ATMPIF_GET_CONFIG
NGM_ATMPIF_GET_LINK_STATUS
NGM_ATMPIF_GET_STATS
NGM_ATMPIF_CLR_STATS
NGM_ATMPIF_GETCLR_STATS

Definition at line 166 of file ng_atmpif.h.

7.10 /usr/src/sys/netgraph/atm/ng_ccatm.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngm_ccatm_addr_req](#)
- struct [ngm_ccatm_get_addresses](#)
- struct [ngm_ccatm_port](#)
- struct [ngm_ccatm_portlist](#)
- struct [ccatm_op](#)

Defines

- #define [NG_CCATM_NODE_TYPE](#) "ccatm"
- #define [NGM_CCATM_COOKIE](#) 984046139
- #define [NGM_CCATM_ADDR_ARRAY_INFO](#)
- #define [NGM_CCATM_UNI_ADDR_INFO](#)
- #define [NGM_CCATM_ADDR_REQ_INFO](#)
- #define [NGM_CCATM_ADDR_REQ_ARRAY_INFO](#)
- #define [NGM_CCATM_GET_ADDRESSES_INFO](#)
- #define [NGM_CCATM_PORT_INFO](#)
- #define [NGM_CCATM_ESI_INFO](#)
- #define [NGM_CCATM_ATM_PORT_INFO](#)
- #define [NGM_CCATM_PORT_ARRAY_INFO](#)
- #define [NGM_CCATM_PORTLIST_INFO](#)

Enumerations

- enum {
 - [NGM_CCATM_DUMP](#), [NGM_CCATM_STOP](#), [NGM_CCATM_START](#), [NGM_CCATM_CLEAR](#),
 - [NGM_CCATM_GET_ADDRESSES](#), [NGM_CCATM_ADDRESS_REGISTERED](#), [NGM_CCATM_ADDRESS_UNREGISTERED](#), [NGM_CCATM_SET_PORT_PARAM](#),
 - [NGM_CCATM_GET_PORT_PARAM](#), [NGM_CCATM_GET_PORTLIST](#), [NGM_CCATM_GETSTATE](#), [NGM_CCATM_SETLOG](#),
 - [NGM_CCATM_RESET](#), [NGM_CCATM_GET_EXSTAT](#) }

7.10.1 Define Documentation

7.10.1.1 #define NG_CCATM_NODE_TYPE "ccatm"

Definition at line 42 of file `ng_ccatm.h`.

7.10.1.2 #define NGM_CCATM_ADDR_ARRAY_INFO**Value:**

```

{
    &ng_parse_hint8_type,
    UNI_ADDR_MAXLEN
}

```

Definition at line 65 of file ng_ccatm.h.

7.10.1.3 #define NGM_CCATM_ADDR_REQ_ARRAY_INFO**Value:**

```

{
    &ng_ccatm_addr_req_type,
    ng_ccatm_addr_req_array_getlen
}

```

Definition at line 101 of file ng_ccatm.h.

7.10.1.4 #define NGM_CCATM_ADDR_REQ_INFO**Value:**

```

{
    { "port", &ng_parse_uint32_type },
    { "addr", &ng_ccatm_uni_addr_type },
    { NULL },
}

```

Definition at line 87 of file ng_ccatm.h.

7.10.1.5 #define NGM_CCATM_ATM_PORT_INFO**Value:**

```

{
    { "port", &ng_parse_uint32_type },
    { "pcr", &ng_parse_uint32_type },
    { "max_vpi_bits", &ng_parse_uint32_type },
    { "max_vci_bits", &ng_parse_uint32_type },
    { "max_svpc_vpi", &ng_parse_uint32_type },
    { "max_svcc_vpi", &ng_parse_uint32_type },
    { "min_svcc_vci", &ng_parse_uint32_type },
    { "esi", &ng_ccatm_esi_type },
    { "num_addr", &ng_parse_uint32_type },
    { NULL }
}

```

Definition at line 134 of file ng_ccatm.h.

7.10.1.6 #define NGM_CCATM_COOKIE 984046139

Definition at line 43 of file ng_ccatm.h.

Referenced by ng_ccatm_rcvmsg().

7.10.1.7 #define NGM_CCATM_ESI_INFO

Value:

```
{
    &ng_parse_hint8_type,
    6
}
```

Definition at line 129 of file ng_ccatm.h.

7.10.1.8 #define NGM_CCATM_GET_ADDRESSES_INFO

Value:

```
{
    { "count", &ng_parse_uint32_type },
    { "addr", &ng_ccatm_addr_req_array_type },
    { NULL }
}
```

Definition at line 106 of file ng_ccatm.h.

7.10.1.9 #define NGM_CCATM_PORT_ARRAY_INFO

Value:

```
{
    &ng_parse_uint32_type,
    ng_ccatm_port_array_getlen
}
```

Definition at line 155 of file ng_ccatm.h.

7.10.1.10 #define NGM_CCATM_PORT_INFO

Value:

```
{
    { "port", &ng_parse_uint32_type },
    { NULL }
}
```

Definition at line 119 of file ng_ccatm.h.

7.10.1.11 #define NGM_CCATM_PORTLIST_INFO**Value:**

```

{
    { "nports",    &ng_parse_uint32_type },      \
    { "ports",    &ng_ccatm_port_array_type },  \
    { NULL }                                          \
}

```

Definition at line 160 of file ng_ccatm.h.

7.10.1.12 #define NGM_CCATM_UNI_ADDR_INFO**Value:**

```

{
    { "type",      &ng_parse_uint32_type },      \
    { "plan",     &ng_parse_uint32_type },      \
    { "len",      &ng_parse_uint32_type },      \
    { "addr",     &ng_ccatm_addr_array_type },  \
    { NULL }                                           \
}

```

Definition at line 71 of file ng_ccatm.h.

7.10.2 Enumeration Type Documentation**7.10.2.1 anonymous enum****Enumerator:**

```

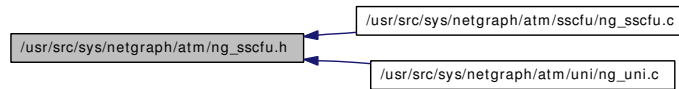
NGM_CCATM_DUMP
NGM_CCATM_STOP
NGM_CCATM_START
NGM_CCATM_CLEAR
NGM_CCATM_GET_ADDRESSES
NGM_CCATM_ADDRESS_REGISTERED
NGM_CCATM_ADDRESS_UNREGISTERED
NGM_CCATM_SET_PORT_PARAM
NGM_CCATM_GET_PORT_PARAM
NGM_CCATM_GET_PORTLIST
NGM_CCATM_GETSTATE
NGM_CCATM_SETLOG
NGM_CCATM_RESET
NGM_CCATM_GET_EXSTAT

```

Definition at line 45 of file ng_ccatm.h.

7.11 /usr/src/sys/netgraph/atm/ng_sscfu.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_sscfu_getdefparam](#)
- struct [sscfu_arg](#)

Defines

- `#define NG_SSCFU_NODE_TYPE "sscfu"`
- `#define NGM_SSCFU_COOKIE 980517963`
- `#define NG_SSCFU_GETDEFPARAM_INFO`

Enumerations

- enum {
 - `NGM_SSCFU_GETDEFPARAM = 1, NGM_SSCFU_ENABLE, NGM_SSCFU_DISABLE,`
 - `NGM_SSCFU_GETDEBUG,`
 - `NGM_SSCFU_SETDEBUG, NGM_SSCFU_GETSTATE }`

7.11.1 Define Documentation

7.11.1.1 `#define NG_SSCFU_GETDEFPARAM_INFO`

Value:

```

{
    { "param",          &ng_sscop_param_type }, \
    { "mask",          &ng_parse_uint32_type }, \
    { NULL }
}
  
```

Definition at line 54 of file `ng_sscfu.h`.

7.11.1.2 `#define NG_SSCFU_NODE_TYPE "sscfu"`

Definition at line 36 of file `ng_sscfu.h`.

7.11.1.3 `#define NGM_SSCFU_COOKIE 980517963`

Definition at line 37 of file `ng_sscfu.h`.

Referenced by `ng_sscfu_rcvmsg()`.

7.11.2 Enumeration Type Documentation

7.11.2.1 anonymous enum

Enumerator:

NGM_SSCFU_GETDEFPARAM

NGM_SSCFU_ENABLE

NGM_SSCFU_DISABLE

NGM_SSCFU_GETDEBUG

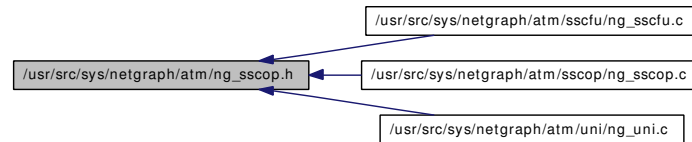
NGM_SSCFU_SETDEBUG

NGM_SSCFU_GETSTATE

Definition at line 40 of file ng_sscfu.h.

7.12 /usr/src/sys/netgraph/atm/ng_sscop.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_sscop_setparam](#)
- struct [ng_sscop_setparam_resp](#)
- struct [sscop_arg](#)
- struct [sscop_marg](#)
- struct [sscop_merr](#)

Defines

- #define [NG_SSCOP_NODE_TYPE](#) "sscop"
- #define [NGM_SSCOP_COOKIE](#) 980175044
- #define [NG_SSCOP_PARAM_INFO](#)
- #define [NG_SSCOP_SETPARAM_INFO](#)
- #define [NG_SSCOP_SETPARAM_RESP_INFO](#)

Enumerations

- enum {
[NGM_SSCOP_GETPARAM](#) = 1, [NGM_SSCOP_SETPARAM](#), [NGM_SSCOP_ENABLE](#), [NGM_SSCOP_DISABLE](#),
[NGM_SSCOP_GETDEBUG](#), [NGM_SSCOP_SETDEBUG](#), [NGM_SSCOP_GETSTATE](#) }

7.12.1 Define Documentation

7.12.1.1 #define NG_SSCOP_NODE_TYPE "sscop"

Definition at line 36 of file `ng_sscop.h`.

7.12.1.2 #define NG_SSCOP_PARAM_INFO

Value:

```

{
    { "timer_cc",          &ng_parse_uint32_type }, \
    { "timer_poll",      &ng_parse_uint32_type }, \
    { "timer_keep_alive", &ng_parse_uint32_type }, \
    { "timer_no_response", &ng_parse_uint32_type }, \

```

```

    { "timer_idle",      &ng_parse_uint32_type }, \
    { "maxk",           &ng_parse_uint32_type }, \
    { "maxj",           &ng_parse_uint32_type }, \
    { "maxcc",          &ng_parse_uint32_type }, \
    { "maxpd",          &ng_parse_uint32_type }, \
    { "maxstat",        &ng_parse_uint32_type }, \
    { "mr",              &ng_parse_uint32_type }, \
    { "flags",          &ng_parse_uint32_type }, \
    { NULL }
}

```

Definition at line 51 of file ng_sscop.h.

7.12.1.3 #define NG_SSCOP_SETPARAM_INFO

Value:

```

{
    { "mask",           &ng_parse_uint32_type }, \
    { "param",         &ng_sscop_param_type }, \
    { NULL }
}

```

Definition at line 73 of file ng_sscop.h.

7.12.1.4 #define NG_SSCOP_SETPARAM_RESP_INFO

Value:

```

{
    { "mask",           &ng_parse_uint32_type }, \
    { "error",         &ng_parse_int32_type }, \
    { NULL }
}

```

Definition at line 84 of file ng_sscop.h.

7.12.1.5 #define NGM_SSCOP_COOKIE 980175044

Definition at line 37 of file ng_sscop.h.

Referenced by ng_sscop_rcvmsg().

7.12.2 Enumeration Type Documentation

7.12.2.1 anonymous enum

Enumerator:

```

    NGM_SSCOP_GETPARAM
    NGM_SSCOP_SETPARAM
    NGM_SSCOP_ENABLE
    NGM_SSCOP_DISABLE

```

NGM_SSCOP_GETDEBUG

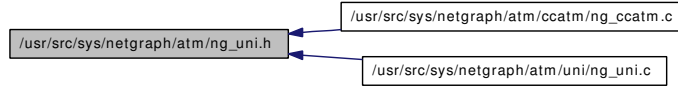
NGM_SSCOP_SETDEBUG

NGM_SSCOP_GETSTATE

Definition at line 40 of file ng_sscop.h.

7.13 /usr/src/sys/netgraph/atm/ng_uni.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngm_uni_debug](#)
- struct [ngm_uni_config_mask](#)
- struct [ngm_uni_set_config](#)
- struct [uni_arg](#)

Defines

- #define [NG_UNI_NODE_TYPE](#) "uni"
- #define [NGM_UNI_COOKIE](#) 981112392
- #define [NGM_UNI_DEBUGLEVEL_INFO](#)
- #define [NGM_UNI_DEBUG_INFO](#)
- #define [NGM_UNI_CONFIG_INFO](#)
- #define [NGM_UNI_CONFIG_MASK_INFO](#)
- #define [NGM_UNI_SET_CONFIG_INFO](#)

Enumerations

- enum {
 - [NGM_UNI_GETDEBUG](#), [NGM_UNI_SETDEBUG](#), [NGM_UNI_GET_CONFIG](#), [NGM_UNI_SET_CONFIG](#),
 - [NGM_UNI_ENABLE](#), [NGM_UNI_DISABLE](#), [NGM_UNI_GETSTATE](#) }

7.13.1 Define Documentation

7.13.1.1 #define NG_UNI_NODE_TYPE "uni"

Definition at line 36 of file ng_uni.h.

7.13.1.2 #define NGM_UNI_CONFIG_INFO

Value:

```

{
    { "proto",    &ng_parse_uint32_type },    \
    { "popt",    &ng_parse_uint32_type },    \
    { "option",  &ng_parse_uint32_type },    \
    { "timer301", &ng_parse_uint32_type },    \
    { "timer303", &ng_parse_uint32_type },    \
}
  
```

```

    { "init303", &ng_parse_uint32_type }, \
    { "timer308", &ng_parse_uint32_type }, \
    { "init308", &ng_parse_uint32_type }, \
    { "timer309", &ng_parse_uint32_type }, \
    { "timer310", &ng_parse_uint32_type }, \
    { "timer313", &ng_parse_uint32_type }, \
    { "timer316", &ng_parse_uint32_type }, \
    { "init316", &ng_parse_uint32_type }, \
    { "timer317", &ng_parse_uint32_type }, \
    { "timer322", &ng_parse_uint32_type }, \
    { "init322", &ng_parse_uint32_type }, \
    { "timer397", &ng_parse_uint32_type }, \
    { "timer398", &ng_parse_uint32_type }, \
    { "timer399", &ng_parse_uint32_type }, \
    { NULL } \
}

```

Definition at line 62 of file ng_uni.h.

7.13.1.3 #define NGM_UNI_CONFIG_MASK_INFO

Value:

```

{
    { "mask", &ng_parse_hint32_type }, \
    { "popt_mask", &ng_parse_hint32_type }, \
    { "option_mask", &ng_parse_hint32_type }, \
    { NULL } \
}

```

Definition at line 91 of file ng_uni.h.

7.13.1.4 #define NGM_UNI_COOKIE 981112392

Definition at line 37 of file ng_uni.h.

Referenced by ng_uni_rcvmsg().

7.13.1.5 #define NGM_UNI_DEBUG_INFO

Value:

```

{
    { "level", &ng_uni_debuglevel_type }, \
    { NULL } \
}

```

Definition at line 56 of file ng_uni.h.

7.13.1.6 #define NGM_UNI_DEBUGLEVEL_INFO

Value:

```

{
    &ng_parse_uint32_type, \
    UNI_MAXFACILITY \
}

```

Definition at line 52 of file ng_uni.h.

7.13.1.7 #define NGM_UNI_SET_CONFIG_INFO

Value:

```
{
    { "config",      &ng_uni_config_type }, \
    { "mask",       &ng_uni_config_mask_type }, \
    { NULL }
}
```

Definition at line 103 of file ng_uni.h.

7.13.2 Enumeration Type Documentation

7.13.2.1 anonymous enum

Enumerator:

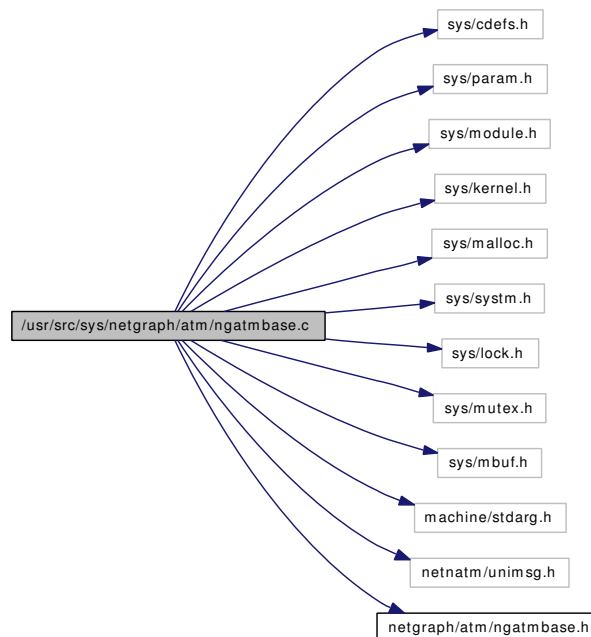
```
NGM_UNI_GETDEBUG
NGM_UNI_SETDEBUG
NGM_UNI_GET_CONFIG
NGM_UNI_SET_CONFIG
NGM_UNI_ENABLE
NGM_UNI_DISABLE
NGM_UNI_GETSTATE
```

Definition at line 39 of file ng_uni.h.

7.14 /usr/src/sys/netgraph/atm/ngatmbase.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/module.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/system.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/mbuf.h>
#include <machine/stdarg.h>
#include <netnatm/unimsg.h>
#include <netgraph/atm/ngatmbase.h>
```

Include dependency graph for ngatmbase.c:



Data Structures

- struct [ngatm_msg](#)

Defines

- #define [NGATMBASE_VERSION](#) 1
- #define [EXTRA](#) 128

Functions

- `__FBSDID` ("FreeBSD: src/sys/netgraph/atm/ngatmbase.c,v 1.3 2005/01/07 01:45:40 imp Exp \$")
- static int `ngatm_handler` (module_t, int, void *)
- `MODULE_VERSION` (ngatmbase, NGATMBASE_VERSION)
- `DECLARE_MODULE` (ngatmbase, `ngatm_data`, SI_SUB_EXEC, SI_ORDER_ANY)
- `MALLOC_DEFINE` (M_UNIMSG, "unimsg", "uni message buffers")
- `MALLOC_DEFINE` (M_UNIMSGHDR, "unimsghdr", "uni message headers")
- static void `uni_msg_init` (void)
- int `uni_msg_extend` (struct uni_msg *m, size_t s)
- int `uni_msg_append` (struct uni_msg *m, void *buf, size_t size)
- mbuf * `uni_msg_pack_mbuf` (struct uni_msg *msg, void *hdr, size_t hdrlen)
- static `LIST_HEAD` (`ngatm_msg`)
- uni_msg * `uni_msg_alloc` (size_t s)
- void `uni_msg_destroy` (struct uni_msg *m)
- uni_msg * `uni_msg_build` (void *ptr,...)
- int `uni_msg_unpack_mbuf` (struct mbuf *m, struct uni_msg **pmsg)

Variables

- static moduledata_t `ngatm_data`
- static struct mtx `ngatm_unilist_mtx`

7.14.1 Define Documentation

7.14.1.1 #define EXTRA 128

Definition at line 67 of file ngatmbase.c.

Referenced by `uni_msg_alloc()`, and `uni_msg_extend()`.

7.14.1.2 #define NGATMBASE_VERSION 1

Definition at line 47 of file ngatmbase.c.

7.14.2 Function Documentation

7.14.2.1 __FBSDID ("FreeBSD: src/sys/netgraph/atm/ngatmbase. c, v 1.3 2005/01/07 01:45:40 imp Exp \$")

7.14.2.2 DECLARE_MODULE (ngatmbase, `ngatm_data`, SI_SUB_EXEC, SI_ORDER_ANY)

7.14.2.3 static LIST_HEAD (`ngatm_msg`) [static]

Definition at line 328 of file ngatmbase.c.

7.14.2.4 `MALLOC_DEFINE (M_UNIMSGHDR, "unimsghdr", "uni message headers")`

7.14.2.5 `MALLOC_DEFINE (M_UNIMSG, "unimsg", "uni message buffers")`

7.14.2.6 `MODULE_VERSION (ngatmbase, NGATMBASE_VERSION)`

7.14.2.7 `static int ngatm_handler (module_t, int, void *) [static]`

Definition at line 481 of file ngatmbase.c.

References `uni_msg_init()`.

Here is the call graph for this function:



7.14.2.8 `struct uni_msg* uni_msg_alloc (size_t s)`

Definition at line 352 of file ngatmbase.c.

References EXTRA.

Referenced by `uni_msg_build()`, and `uni_msg_unpack_mbuf()`.

7.14.2.9 `int uni_msg_append (struct uni_msg * m, void * buf, size_t size)`

Definition at line 118 of file ngatmbase.c.

7.14.2.10 `struct uni_msg* uni_msg_build (void * ptr, ...)`

Definition at line 411 of file ngatmbase.c.

References `uni_msg_alloc()`.

Here is the call graph for this function:



7.14.2.11 `void uni_msg_destroy (struct uni_msg * m)`

Definition at line 387 of file ngatmbase.c.

Referenced by `ng_ccatm_rcvdata()`, `ng_ccatm_rcvuni()`, `ng_ccatm_send_uni_glob()`, `ng_ccatm_send_uni_glob()`, `ng_uni_rcvlower()`, `ng_uni_rcvupper()`, `uni_msg_extend()`, `uni_saal_output()`, and `uni_uni_output()`.

7.14.2.12 `int uni_msg_extend (struct uni_msg * m, size_t s)`

Definition at line 89 of file ngatmbase.c.

References EXTRA, and uni_msg_destroy().

Here is the call graph for this function:



7.14.2.13 static void uni_msg_init (void) [static]

Definition at line 76 of file ngatmbase.c.

References ngatm_unilist_mtx.

Referenced by ngatm_handler().

7.14.2.14 struct mbuf* uni_msg_pack_mbuf (struct uni_msg * msg, void * hdr, size_t hdrLEN)

Definition at line 136 of file ngatmbase.c.

Referenced by ng_ccatm_send_uni(), ng_ccatm_send_uni_glob(), uni_saal_output(), and uni_uni_output().

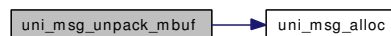
7.14.2.15 int uni_msg_unpack_mbuf (struct mbuf * m, struct uni_msg ** pmsg)

Definition at line 458 of file ngatmbase.c.

References uni_msg_alloc().

Referenced by ng_ccatm_rcvdata(), ng_ccatm_rcvuni(), ng_uni_rcvlower(), and ng_uni_rcvupper().

Here is the call graph for this function:



7.14.3 Variable Documentation

7.14.3.1 moduledata_t ngatm_data [static]

Initial value:

```

{
    "ngatmbase",
    ngatm_handler,
    0
}
  
```

Definition at line 51 of file ngatmbase.c.

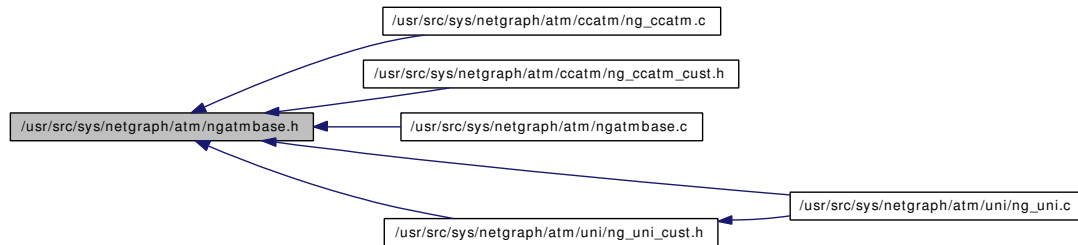
7.14.3.2 struct mtx ngatm_unilist_mtx [static]

Definition at line 70 of file ngatmbase.c.

Referenced by uni_msg_init().

7.15 /usr/src/sys/netgraph/atm/ngatmbase.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- mbuf * [uni_msg_pack_mbuf](#) (struct uni_msg *, void *, size_t)
- uni_msg * [uni_msg_alloc](#) (size_t)
- uni_msg * [uni_msg_build](#) (void *,...)
- void [uni_msg_destroy](#) (struct uni_msg *)
- int [uni_msg_unpack_mbuf](#) (struct mbuf *, struct uni_msg **)

7.15.1 Function Documentation

7.15.1.1 struct uni_msg* uni_msg_alloc (size_t)

Definition at line 352 of file ngatmbase.c.

References EXTRA.

Referenced by [uni_msg_build](#)(), and [uni_msg_unpack_mbuf](#)().

7.15.1.2 struct uni_msg* uni_msg_build (void *, ...)

Definition at line 411 of file ngatmbase.c.

References [uni_msg_alloc](#)().

Here is the call graph for this function:



7.15.1.3 void uni_msg_destroy (struct uni_msg *)

Definition at line 387 of file ngatmbase.c.

Referenced by [ng_ccatm_rcvdata](#)(), [ng_ccatm_rcvuni](#)(), [ng_ccatm_send_uni](#)(), [ng_ccatm_send_uni_glob](#)(), [ng_uni_rcvlower](#)(), [ng_uni_rcvupper](#)(), [uni_msg_extend](#)(), [uni_saal_output](#)(), and [uni_uni_output](#)().

7.15.1.4 struct mbuf* uni_msg_pack_mbuf (struct uni_msg *, void *, size_t)

Definition at line 136 of file ngatmbase.c.

Referenced by `ng_ccatm_send_uni()`, `ng_ccatm_send_uni_glob()`, `uni_saal_output()`, and `uni_uni_output()`.

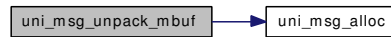
7.15.1.5 int uni_msg_unpack_mbuf (struct mbuf *, struct uni_msg **)

Definition at line 458 of file ngatmbase.c.

References `uni_msg_alloc()`.

Referenced by `ng_ccatm_rcvdata()`, `ng_ccatm_rcvuni()`, `ng_uni_rcvlower()`, and `ng_uni_rcvupper()`.

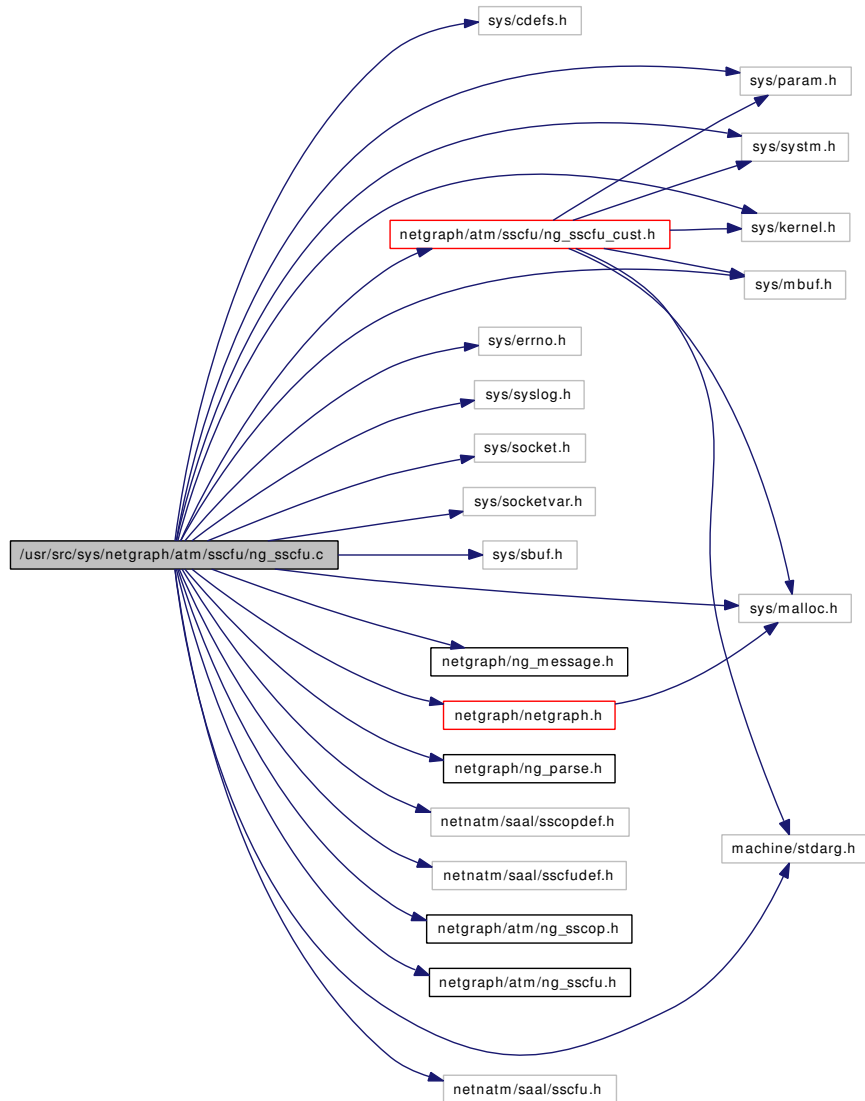
Here is the call graph for this function:



7.16 /usr/src/sys/netgraph/atm/sscfu/ng_sscfu.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sbuf.h>
#include <machine/stdarg.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netnatm/saal/sscopdef.h>
#include <netnatm/saal/sscfudef.h>
#include <netgraph/atm/ng_sscop.h>
#include <netgraph/atm/ng_sscfu.h>
#include <netgraph/atm/sscfu/ng_sscfu_cust.h>
#include <netnatm/saal/sscfu.h>
```

Include dependency graph for ng_sscfu.c:



Data Structures

- struct [priv](#)

Functions

- [__FBSDID](#) ("FreeBSD: src/sys/netgraph/atm/sscfu/ng_sscfu.c,v 1.4 2005/01/07 01:45:41 imp Exp \$")
- [MALLOC_DEFINE](#) (M_NG_SSCFU,"netgraph_sscfu","netgraph uni sscf node")
- [MODULE_DEPEND](#) (ng_sscfu, ngatmbase, 1, 1, 1)
- static int [ng_sscfu_mod_event](#) (module_t, int, void *)
- [NETGRAPH_INIT](#) (sscfu,&[ng_sscfu_tpestruct](#))
- static void [sscfu_send_upper](#) (struct sscfu *, void *, enum saal_sig, struct mbuf *)
- static void [sscfu_send_lower](#) (struct sscfu *, void *, enum sscop_aasig, struct mbuf *, u_int)
- static void [sscfu_window](#) (struct sscfu *, void *, u_int)

- static void `sscfu_verbose` (struct `sscfu *`, void *, const char *,...)
- static int `text_status` (node_p node, struct `priv *priv`, char *arg, u_int len)
- static int `ng_sscfu_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_sscfu_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_sscfu_disconnect` (hook_p hook)
- static int `ng_sscfu_rcvupper` (hook_p hook, item_p item)
- static int `ng_sscfu_rcvlower` (hook_p hook, item_p item)
- static int `ng_sscfu_constructor` (node_p node)
- static int `ng_sscfu_shutdown` (node_p node)

Variables

- static struct `ng_parse_struct_field` `ng_sscop_param_type_info` []
- static struct `ng_parse_type` `ng_sscop_param_type`
- static struct `ng_parse_struct_field` `ng_sscfu_getdefparam_type_info` []
- static struct `ng_parse_type` `ng_sscfu_getdefparam_type`
- static struct `ng_cmdlist` `ng_sscfu_cmdlist` []
- static `ng_constructor_t` `ng_sscfu_constructor`
- static `ng_shutdown_t` `ng_sscfu_shutdown`
- static `ng_rcvmsg_t` `ng_sscfu_rcvmsg`
- static `ng_newhook_t` `ng_sscfu_newhook`
- static `ng_disconnect_t` `ng_sscfu_disconnect`
- static `ng_rcvdata_t` `ng_sscfu_rcvupper`
- static `ng_rcvdata_t` `ng_sscfu_rcvlower`
- static struct `ng_type` `ng_sscfu_typestruct`

7.16.1 Function Documentation

7.16.1.1 `__FBSDID` ("\$FreeBSD: src/sys/netgraph/atm/sscfu/ng_sscfu.c, v 1.4 2005/01/07 01:45:41 imp Exp \$")

7.16.1.2 `MALLOC_DEFINE` (M_NG_SSCFU, "netgraph_sscfu", "netgraph uni sscf node")

7.16.1.3 `MODULE_DEPEND` (ng_sscfu, ngatmbase, 1, 1, 1)

7.16.1.4 `NETGRAPH_INIT` (sscfu, & `ng_sscfu_typestruct`)

7.16.1.5 `static int ng_sscfu_constructor` (node_p node) [static]

Definition at line 537 of file `ng_sscfu.c`.

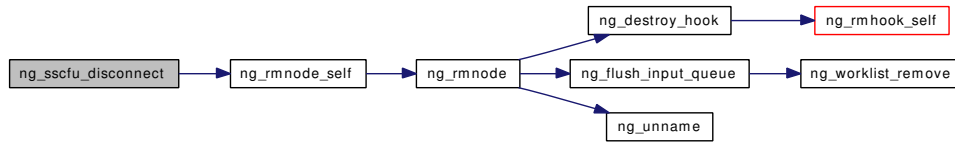
References `NG_NODE_SET_PRIVATE`.

7.16.1.6 `static int ng_sscfu_disconnect` (hook_p hook) [static]

Definition at line 364 of file `ng_sscfu.c`.

References `priv::enabled`, `priv::lower`, `NG_HOOK_NODE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, `ng_rmnode_self()`, `priv::sscf`, and `priv::upper`.

Here is the call graph for this function:



7.16.1.7 `static int ng_sscfu_mod_event (module_t, int, void *)` [static]

Definition at line 589 of file `ng_sscfu.c`.

7.16.1.8 `static int ng_sscfu_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 349 of file `ng_sscfu.c`.

References `priv::lower`, `NG_HOOK_SET_RCVDATA`, `NG_NODE_PRIVATE`, `ng_sscfu_rcvlower`, and `priv::upper`.

7.16.1.9 `static int ng_sscfu_rcvlower (hook_p hook, item_p item)` [static]

Definition at line 458 of file `ng_sscfu.c`.

References `priv::enabled`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `priv::sscf`, and `priv::upper`.

7.16.1.10 `static int ng_sscfu_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 215 of file `ng_sscfu.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `priv::enabled`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NG_TEXTRESPONSE`, `NGI_GET_MSG`, `NGM_GENERIC_COOKIE`, `NGM_SSCFU_COOKIE`, `NGM_SSCFU_DISABLE`, `NGM_SSCFU_ENABLE`, `NGM_SSCFU_GETDEBUG`, `NGM_SSCFU_GETDEFPARAM`, `NGM_SSCFU_GETSTATE`, `NGM_SSCFU_SETDEBUG`, `NGM_TEXT_STATUS`, `priv::sscf`, `text_status()`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.16.1.11 `static int ng_sscfu_rcvupper (hook_p hook, item_p item)` [static]

Definition at line 399 of file `ng_sscfu.c`.

References `priv::enabled`, `priv::lower`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `priv::sscf`.

7.16.1.12 `static int ng_sscfu_shutdown (node_p node)` [static]

Definition at line 555 of file ng_sscfu.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and priv::sscf.

7.16.1.13 `static void sscfu_send_lower (struct sscfu *, void *, enum sscop_aasig, struct mbuf *, u_int)` [static]

Definition at line 493 of file ng_sscfu.c.

References sscop_arg::arg, priv::lower, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, and sscop_arg::sig.

Referenced by sscfu_verbose().

7.16.1.14 `static void sscfu_send_upper (struct sscfu *, void *, enum saal_sig, struct mbuf *)` [static]

Definition at line 428 of file ng_sscfu.c.

References NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, sscfu_arg::sig, and priv::upper.

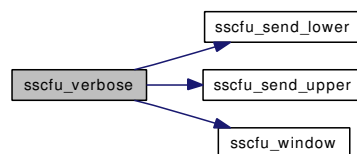
Referenced by sscfu_verbose().

7.16.1.15 `static void sscfu_verbose (struct sscfu * sscfu, void * arg, const char * fmt, ...)` [static]

Definition at line 169 of file ng_sscfu.c.

References sscfu_send_lower(), sscfu_send_upper(), and sscfu_window().

Here is the call graph for this function:

**7.16.1.16** `static void sscfu_window (struct sscfu *, void *, u_int)` [static]

Definition at line 528 of file ng_sscfu.c.

Referenced by sscfu_verbose().

7.16.1.17 `static int text_status (node_p node, struct priv * priv, char * arg, u_int len)` [static]

Definition at line 184 of file ng_sscfu.c.

References priv::enabled, priv::lower, NG_HOOK_NAME, NG_HOOK_NODE, NG_HOOK_PEER, NG_NODE_NAME, priv::sscf, and priv::upper.

7.16.2 Variable Documentation

7.16.2.1 `struct ng_cmdlist ng_sscfu_cmdlist[]` [static]

Definition at line 94 of file `ng_sscfu.c`.

7.16.2.2 `ng_constructor_t ng_sscfu_constructor` [static]

Definition at line 140 of file `ng_sscfu.c`.

7.16.2.3 `ng_disconnect_t ng_sscfu_disconnect` [static]

Definition at line 144 of file `ng_sscfu.c`.

7.16.2.4 `struct ng_parse_type ng_sscfu_getdefparam_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_sscfu_getdefparam_type_info
}
```

Definition at line 88 of file `ng_sscfu.c`.

7.16.2.5 `struct ng_parse_struct_field ng_sscfu_getdefparam_type_info[]` [static]

Initial value:

```
NG_SSCFU_GETDEFPARAM_INFO
```

Definition at line 85 of file `ng_sscfu.c`.

7.16.2.6 `ng_newhook_t ng_sscfu_newhook` [static]

Definition at line 143 of file `ng_sscfu.c`.

7.16.2.7 `ng_rcvdata_t ng_sscfu_rcvlower` [static]

Definition at line 146 of file `ng_sscfu.c`.

Referenced by `ng_sscfu_newhook()`.

7.16.2.8 `ng_rcvmsg_t ng_sscfu_rcvmsg` [static]

Definition at line 142 of file `ng_sscfu.c`.

7.16.2.9 `ng_rcvdata_t ng_sscfu_rcvupper` [static]

Definition at line 145 of file `ng_sscfu.c`.

7.16.2.10 `ng_shutdown_t ng_sscfu_shutdown` [static]

Definition at line 141 of file ng_sscfu.c.

7.16.2.11 `struct ng_type ng_sscfu_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_SSCFU_NODE_TYPE,
    .mod_event =    ng_sscfu_mod_event,
    .constructor = ng_sscfu_constructor,
    .rcvmsg =       ng_sscfu_rcvmsg,
    .shutdown =     ng_sscfu_shutdown,
    .newhook =      ng_sscfu_newhook,
    .rcvdata =      ng_sscfu_rcvupper,
    .disconnect =  ng_sscfu_disconnect,
    .cmdlist =      ng_sscfu_cmdlist,
}
```

Definition at line 150 of file ng_sscfu.c.

7.16.2.12 `struct ng_parse_type ng_sscop_param_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_sscop_param_type_info
}
```

Definition at line 80 of file ng_sscfu.c.

7.16.2.13 `struct ng_parse_struct_field ng_sscop_param_type_info[]` [static]**Initial value:**

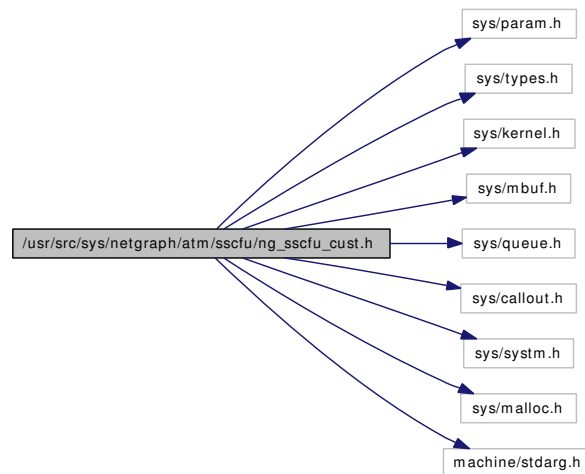
```
NG_SSCOP_PARAM_INFO
```

Definition at line 77 of file ng_sscfu.c.

7.17 /usr/src/sys/netgraph/atm/sscfu/ng_sscfu_cust.h File Reference

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <sys/callout.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <machine/stdarg.h>
```

Include dependency graph for ng_sscfu_cust.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define [MEMINIT\(\)](#)
- #define [MEMZALLOC\(PTR, CAST, SIZE\)](#) ((PTR) = (CAST)malloc((SIZE), M_NG_SSCFU, M_NOWAIT | M_ZERO))
- #define [MEMFREE\(PTR\)](#) free(PTR, M_NG_SSCFU)
- #define [SIG_ALLOC\(PTR\)](#) MEMZALLOC(PTR, struct sscfu_sig *, sizeof(struct sscfu_sig))
- #define [SIG_FREE\(PTR\)](#) MEMFREE(PTR)
- #define [SIGQ_INIT\(Q\)](#) TAILQ_INIT(Q)
- #define [SIGQ_APPEND\(Q, S\)](#) TAILQ_INSERT_TAIL(Q, S, link)
- #define [SIGQ_GET\(Q\)](#) ng_sscfu_sigq_get((Q))
- #define [DECL_SIGQ_GET](#)

- #define [SIGQ_CLEAR\(Q\)](#)
- #define [MBUF_FREE\(M\) m_freem\(M\)](#)
- #define [ASSERT\(S\)](#)

Functions

- typedef [TAILQ_ENTRY](#) (sscfu_sig) sscfu_sigq_link_t
- typedef [TAILQ_HEAD](#) (sscfu_sigq, sscfu_sig) sscfu_sigq_head_t

7.17.1 Define Documentation

7.17.1.1 #define ASSERT(S)

Definition at line 130 of file ng_sscfu_cust.h.

7.17.1.2 #define DECL_SIGQ_GET

Value:

```
static __inline struct sscfu_sig *
ng_sscfu_sigq_get(struct sscfu_sigq *q)
{
    struct sscfu_sig *s;
    s = TAILQ_FIRST(q);
    if (s != NULL)
        TAILQ_REMOVE(q, s, link);
    return (s);
}
```

Definition at line 94 of file ng_sscfu_cust.h.

7.17.1.3 #define MBUF_FREE(M) m_freem(M)

Definition at line 125 of file ng_sscfu_cust.h.

7.17.1.4 #define MEMFREE(PTR) free(PTR, M_NG_SSCFU)

Definition at line 55 of file ng_sscfu_cust.h.

7.17.1.5 #define MEMINIT()

Value:

```
MALLOC_DECLARE (M_NG_SSCFU); \
DECL_SIGQ_GET
```

Definition at line 49 of file ng_sscfu_cust.h.

7.17.1.6 #define MEMZALLOC(PTR, CAST, SIZE) ((PTR) = (CAST)malloc((SIZE), M_NG_SSCFU, M_NOWAIT | M_ZERO))

Definition at line 53 of file ng_sscfu_cust.h.

7.17.1.7 #define SIG_ALLOC(PTR) MEMZALLOC(PTR, struct sscfu_sig *, sizeof(struct sscfu_sig))

Definition at line 58 of file ng_sscfu_cust.h.

7.17.1.8 #define SIG_FREE(PTR) MEMFREE(PTR)

Definition at line 60 of file ng_sscfu_cust.h.

7.17.1.9 #define SIGQ_APPEND(Q, S) TAILQ_INSERT_TAIL(Q, S, link)

Definition at line 90 of file ng_sscfu_cust.h.

7.17.1.10 #define SIGQ_CLEAR(Q)

Value:

```
do {
    struct sscfu_sig *_s1, *_s2;
    _s1 = TAILQ_FIRST(Q);
    while (_s1 != NULL) {
        _s2 = TAILQ_NEXT(_s1, link);
        if (_s1->m)
            MBUF_FREE(_s1->m);
        SIG_FREE(_s1);
        _s1 = _s2;
    }
    TAILQ_INIT(Q);
} while (0)
```

Definition at line 106 of file ng_sscfu_cust.h.

7.17.1.11 #define SIGQ_GET(Q) ng_sscfu_sigq_get((Q))

Definition at line 92 of file ng_sscfu_cust.h.

7.17.1.12 #define SIGQ_INIT(Q) TAILQ_INIT(Q)

Definition at line 89 of file ng_sscfu_cust.h.

7.17.2 Function Documentation

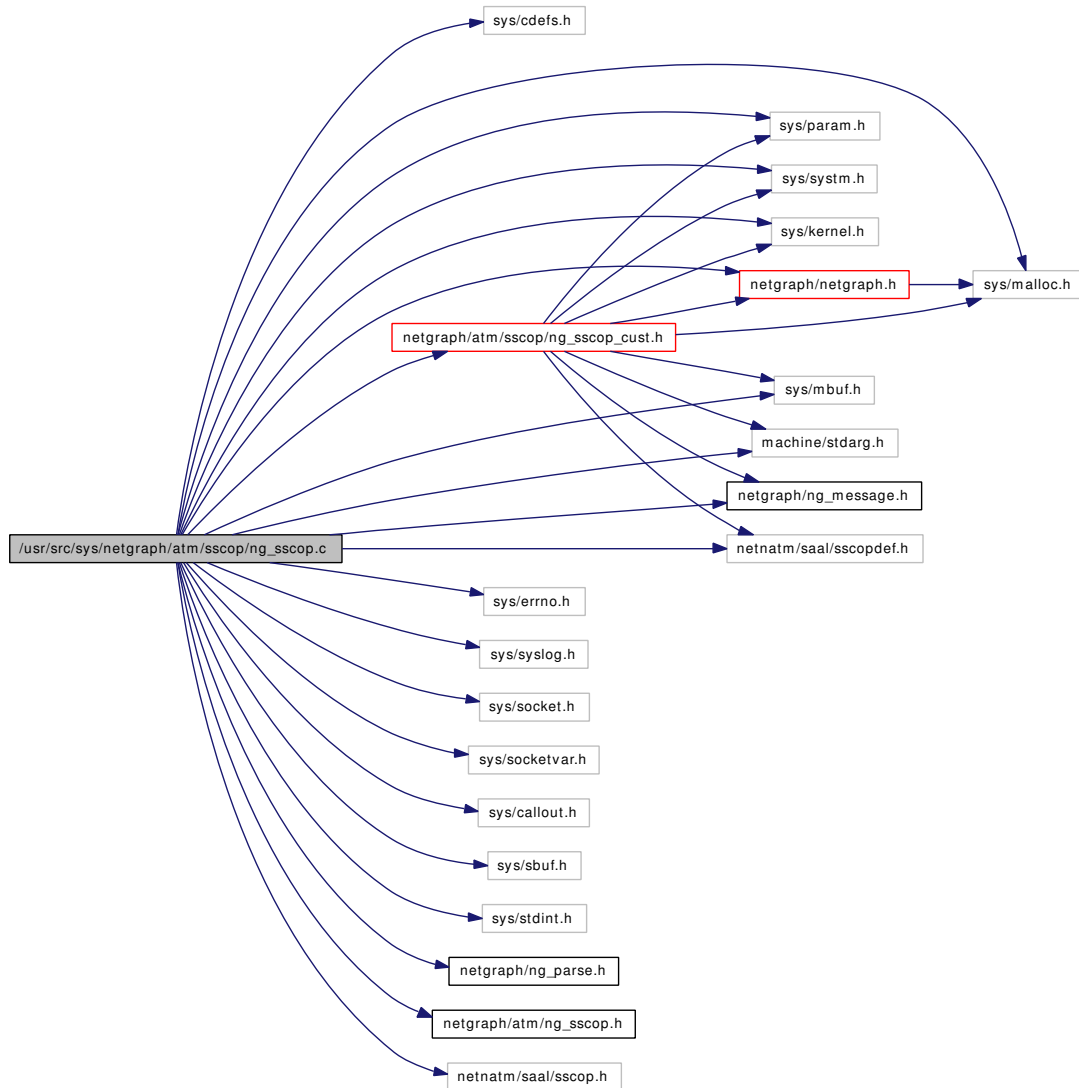
7.17.2.1 typedef TAILQ_ENTRY (sscfu_sig)

7.17.2.2 typedef TAILQ_HEAD (sscfu_sigq, sscfu_sig)

7.18 /usr/src/sys/netgraph/atm/sscop/ng_sscop.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/callout.h>
#include <sys/sbuf.h>
#include <sys/stdint.h>
#include <machine/stdarg.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netnatm/saal/sscopdef.h>
#include <netgraph/atm/ng_sscop.h>
#include <netgraph/atm/sscop/ng_sscop_cust.h>
#include <netnatm/saal/sscop.h>
```

Include dependency graph for ng_sscop.c:



Data Structures

- struct [stats](#)
- struct [priv](#)

Defines

- #define [DDD](#) printf("%s: %d\n", __func__, __LINE__)
- #define [VERBOSE](#)(P, M, F)

Functions

- [__FBSID](#) ("FreeBSD: src/sys/netgraph/atm/sscop/ng_sscop.c,v 1.4 2005/08/10 06:25:40 obrien Exp \$")
- [MALLOC_DEFINE](#)(M_NG_SSCOP,"netgraph_sscop","netgraph sscop node")

- `MODULE_DEPEND` (`ng_sscop`, `ngatmbase`, 1, 1, 1)
- static int `ng_sscop_mod_event` (`module_t`, int, void *)
- `NETGRAPH_INIT` (`sscop`, &`ng_sscop_tpestruct`)
- static void `sscop_send_manage` (`struct sscop *`, void *, enum `sscop_maasig`, `struct SSCOP_MBUF_T *`, u_int, u_int)
- static void `sscop_send_upper` (`struct sscop *`, void *, enum `sscop_aasig`, `struct SSCOP_MBUF_T *`, u_int)
- static void `sscop_send_lower` (`struct sscop *`, void *, `struct SSCOP_MBUF_T *`)
- static void `sscop_verbose` (`struct sscop *`, void *, const char *,...)
- static int `ng_sscop_constructor` (`node_p` node)
- static int `ng_sscop_shutdown` (`node_p` node)
- static int `flow_upper` (`node_p` node, `struct ng_mesg *msg`)
- static int `flow_lower` (`node_p` node, `struct ng_mesg *msg`)
- static int `text_status` (`node_p` node, `struct priv *priv`, char *arg, u_int len)
- static int `ng_sscop_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_sscop_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_sscop_disconnect` (`hook_p` hook)
- static int `ng_sscop_rcvlower` (`hook_p` hook, `item_p` item)
- static void `sscop_send_lower` (`struct sscop *sscop`, void *p, `struct mbuf *m`)
- static int `ng_sscop_rcvupper` (`hook_p` hook, `item_p` item)
- static int `ng_sscop_rcvmanage` (`hook_p` hook, `item_p` item)

Variables

- static struct `ng_parse_struct_field` `ng_sscop_param_type_info` []
- static struct `ng_parse_type` `ng_sscop_param_type`
- static struct `ng_parse_struct_field` `ng_sscop_setparam_type_info` []
- static struct `ng_parse_type` `ng_sscop_setparam_type`
- static struct `ng_parse_struct_field` `ng_sscop_setparam_resp_type_info` []
- static struct `ng_parse_type` `ng_sscop_setparam_resp_type`
- static struct `ng_cmdlist` `ng_sscop_cmdlist` []
- static `ng_constructor_t` `ng_sscop_constructor`
- static `ng_shutdown_t` `ng_sscop_shutdown`
- static `ng_rcvmsg_t` `ng_sscop_rcvmsg`
- static `ng_newhook_t` `ng_sscop_newhook`
- static `ng_disconnect_t` `ng_sscop_disconnect`
- static `ng_rcvdata_t` `ng_sscop_rcvlower`
- static `ng_rcvdata_t` `ng_sscop_rcvupper`
- static `ng_rcvdata_t` `ng_sscop_rcvmanage`
- static struct `ng_type` `ng_sscop_tpestruct`

7.18.1 Define Documentation

7.18.1.1 #define DDD printf("%s: %d\n", __func__, __LINE__)

Definition at line 58 of file `ng_sscop.c`.

7.18.1.2 #define VERBOSE(P, M, F)

Definition at line 67 of file ng_sscop.c.

Referenced by flow_upper().

7.18.2 Function Documentation**7.18.2.1 __FBSDID ("\$FreeBSD: src/sys/netgraph/atm/sscop/ng_sscop. c, v 1.4 2005/08/10 06:25:40 obrien Exp \$")****7.18.2.2 static int flow_lower (node_p node, struct ng_mesg * msg) [static]**

Definition at line 351 of file ng_sscop.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::header, NG_NODE_PRIVATE, NGM_HIGH_WATER_PASSED, NGM_LOW_WATER_PASSED, and priv::sscop.

Referenced by ng_sscop_rcvmsg().

7.18.2.3 static int flow_upper (node_p node, struct ng_mesg * msg) [static]

Definition at line 295 of file ng_sscop.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ngm_queue_state::current, ng_mesg::data, priv::flow, ng_mesg::header, ngm_queue_state::high_watermark, ngm_queue_state::max_queuelen_packets, NG_NODE_PRIVATE, NGM_HIGH_WATER_PASSED, NGM_LOW_WATER_PASSED, NGM_SYNC_QUEUE_STATE, priv::sscop, and VERBOSE.

Referenced by ng_sscop_rcvmsg().

7.18.2.4 MALLOC_DEFINE (M_NG_SSCOP, "netgraph_sscop", "netgraph sscop node")**7.18.2.5 MODULE_DEPEND (ng_sscop, ngatmbase, 1, 1, 1)****7.18.2.6 NETGRAPH_INIT (sscop, & ng_sscop_typestruct)****7.18.2.7 static int ng_sscop_constructor (node_p node) [static]**

Definition at line 245 of file ng_sscop.c.

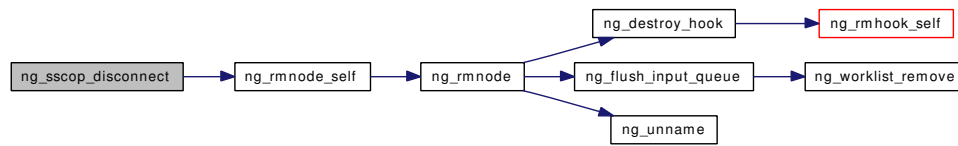
References NG_NODE_FORCE_WRITER, and NG_NODE_SET_PRIVATE.

7.18.2.8 static int ng_sscop_disconnect (hook_p hook) [static]

Definition at line 630 of file ng_sscop.c.

References priv::enabled, priv::lower, priv::manage, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), priv::sscop, and priv::upper.

Here is the call graph for this function:



7.18.2.9 `static int ng_sscop_mod_event (module_t, int, void *)` [static]

Definition at line 863 of file `ng_sscop.c`.

7.18.2.10 `static int ng_sscop_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 612 of file `ng_sscop.c`.

References `priv::lower`, `priv::manage`, `NG_HOOK_SET_RCVDATA`, `NG_NODE_PRIVATE`, `ng_sscop_rcvmanage`, `ng_sscop_rcvupper`, and `priv::upper`.

7.18.2.11 `static int ng_sscop_rcvlower (hook_p hook, item_p item)` [static]

Definition at line 665 of file `ng_sscop.c`.

References `priv::enabled`, `stats::in_dropped`, `stats::in_packets`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `priv::sscop`, `priv::stats`, and `priv::upper`.

7.18.2.12 `static int ng_sscop_rcvmanage (hook_p hook, item_p item)` [static]

Definition at line 783 of file `ng_sscop.c`.

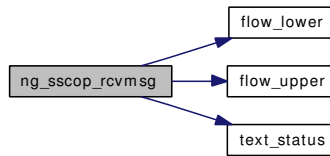
References `priv::enabled`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `priv::sscop`.

7.18.2.13 `static int ng_sscop_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 444 of file `ng_sscop.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `priv::enabled`, `priv::flow`, `flow_lower()`, `flow_upper()`, `ng_mesg::header`, `priv::lower`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NG_TEXTRESPONSE`, `NGI_GET_MSG`, `NGM_FLOW_COOKIE`, `NGM_GENERIC_COOKIE`, `NGM_SSCOP_COOKIE`, `NGM_SSCOP_DISABLE`, `NGM_SSCOP_ENABLE`, `NGM_SSCOP_GETDEBUG`, `NGM_SSCOP_GETPARAM`, `NGM_SSCOP_GETSTATE`, `NGM_SSCOP_SETDEBUG`, `NGM_SSCOP_SETPARAM`, `NGM_TEXT_STATUS`, `priv::sscop`, `priv::stats`, `text_status()`, `ng_mesg::ng_msghdr::typecookie`, and `priv::upper`.

Here is the call graph for this function:



7.18.2.14 `static int ng_sscop_rcvupper (hook_p hook, item_p item)` [static]

Definition at line 709 of file `ng_sscop.c`.

References `priv::enabled`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `priv::sscop`.

7.18.2.15 `static int ng_sscop_shutdown (node_p node)` [static]

Definition at line 265 of file `ng_sscop.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `priv::sscop`.

7.18.2.16 `static void sscop_send_lower (struct sscop * sscop, void * p, struct mbuf * m)` [static]

Definition at line 692 of file `ng_sscop.c`.

References `priv::lower`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `stats::out_dropped`, `stats::out_packets`, and `priv::stats`.

7.18.2.17 `static void sscop_send_lower (struct sscop *, void *, struct SSCOP_MBUF_T *)` [static]

Referenced by `sscop_verbose()`.

7.18.2.18 `static void sscop_send_manage (struct sscop *, void *, enum sscop_maasig, struct SSCOP_MBUF_T *, u_int, u_int)` [static]

Definition at line 806 of file `ng_sscop.c`.

References `sscop_merr::cnt`, `sscop_merr::err`, `stats::errors`, `stats::maa_dropped`, `stats::maa_signals`, `priv::manage`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `sscop_marg::sig`, `sscop_merr::sig`, and `priv::stats`.

Referenced by `sscop_verbose()`.

7.18.2.19 `static void sscop_send_upper (struct sscop *, void *, enum sscop_aasig, struct SSCOP_MBUF_T *, u_int)` [static]

Definition at line 742 of file `ng_sscop.c`.

References `stats::aa_dropped`, `stats::aa_signals`, `sscop_arg::arg`, `stats::data_delivered`, `priv::flow`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `sscop_arg::sig`, `priv::sscop`, `priv::stats`, and `priv::upper`.

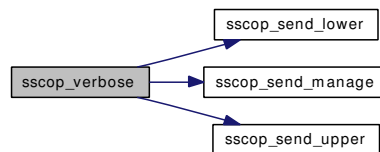
Referenced by sscop_verbose().

7.18.2.20 `static void sscop_verbose (struct sscop * sscop, void * arg, const char * fmt, ...)`
`[static]`

Definition at line 218 of file ng_sscop.c.

References sscop_send_lower(), sscop_send_manage(), and sscop_send_upper().

Here is the call graph for this function:



7.18.2.21 `static int text_status (node_p node, struct priv * priv, char * arg, u_int len)` `[static]`

Definition at line 378 of file ng_sscop.c.

References stats::aa_dropped, stats::aa_signals, stats::data_delivered, priv::enabled, stats::errors, stats::in_dropped, stats::in_packets, priv::lower, stats::maa_dropped, stats::maa_signals, priv::manage, NG_HOOK_NAME, NG_HOOK_NODE, NG_HOOK_PEER, NG_NODE_NAME, stats::out_dropped, stats::out_packets, priv::sscop, priv::stats, and priv::upper.

7.18.3 Variable Documentation

7.18.3.1 `struct ng_cmdlist ng_sscop_cmdlist[]` `[static]`

Definition at line 134 of file ng_sscop.c.

7.18.3.2 `ng_constructor_t ng_sscop_constructor` `[static]`

Definition at line 187 of file ng_sscop.c.

7.18.3.3 `ng_disconnect_t ng_sscop_disconnect` `[static]`

Definition at line 191 of file ng_sscop.c.

7.18.3.4 `ng_newhook_t ng_sscop_newhook` `[static]`

Definition at line 190 of file ng_sscop.c.

7.18.3.5 `struct ng_parse_type ng_sscop_param_type` `[static]`

Initial value:

```
{
    &ng_parse_struct_type,
    ng_sscop_param_type_info
}
```

Definition at line 107 of file ng_sscop.c.

7.18.3.6 struct [ng_parse_struct_field ng_sscop_param_type_info](#)[] [static]

Initial value:

```
NG_SSCOP_PARAM_INFO
```

Definition at line 104 of file ng_sscop.c.

7.18.3.7 [ng_rcvdata_t ng_sscop_rcvlower](#) [static]

Definition at line 192 of file ng_sscop.c.

7.18.3.8 [ng_rcvdata_t ng_sscop_rcvmanage](#) [static]

Definition at line 194 of file ng_sscop.c.

Referenced by ng_sscop_newhook().

7.18.3.9 [ng_rcvmsg_t ng_sscop_rcvmsg](#) [static]

Definition at line 189 of file ng_sscop.c.

7.18.3.10 [ng_rcvdata_t ng_sscop_rcvupper](#) [static]

Definition at line 193 of file ng_sscop.c.

Referenced by ng_sscop_newhook().

7.18.3.11 struct [ng_parse_type ng_sscop_setparam_resp_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_sscop_setparam_resp_type_info,
}
```

Definition at line 129 of file ng_sscop.c.

7.18.3.12 struct [ng_parse_struct_field ng_sscop_setparam_resp_type_info](#)[] [static]

Initial value:

```
NG_SSCOP_SETPARAM_RESP_INFO
```

Definition at line 126 of file ng_sscop.c.

7.18.3.13 struct [ng_parse_type](#) [ng_sscop_setparam_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_sscop_setparam_type_info,
}
```

Definition at line 118 of file ng_sscop.c.

7.18.3.14 struct [ng_parse_struct_field](#) [ng_sscop_setparam_type_info](#)[] [static]**Initial value:**

```
NG_SSCOP_SETPARAM_INFO
```

Definition at line 115 of file ng_sscop.c.

7.18.3.15 [ng_shutdown_t](#) [ng_sscop_shutdown](#) [static]

Definition at line 188 of file ng_sscop.c.

7.18.3.16 struct [ng_type](#) [ng_sscop_tpestruct](#) [static]**Initial value:**

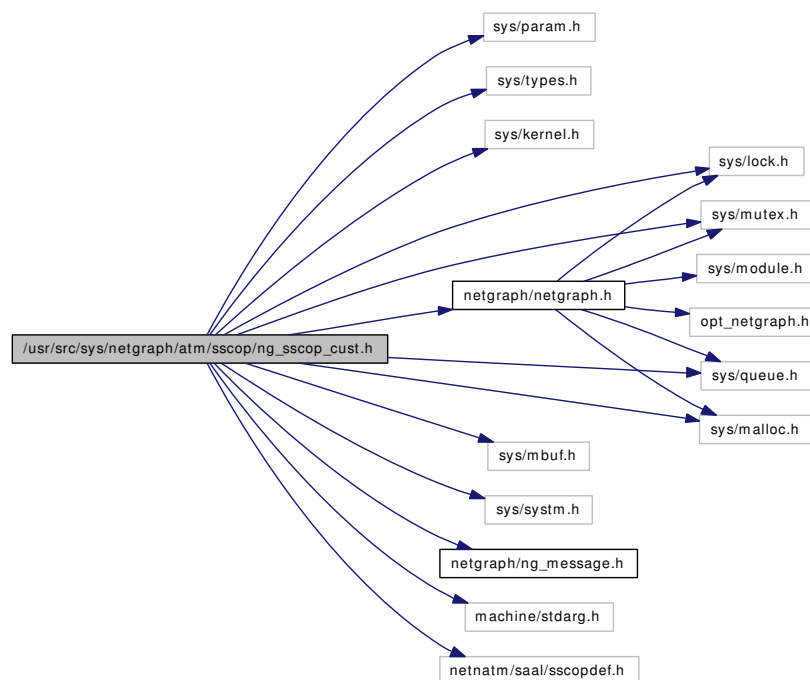
```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_SSCOP_NODE_TYPE,
    .mod_event =   ng_sscop_mod_event,
    .constructor = ng_sscop_constructor,
    .rcvmsg =      ng_sscop_rcvmsg,
    .shutdown =    ng_sscop_shutdown,
    .newhook =     ng_sscop_newhook,
    .rcvdata =     ng_sscop_rcvlower,
    .disconnect =  ng_sscop_disconnect,
    .cmdlist =     ng_sscop_cmdlist,
}
```

Definition at line 198 of file ng_sscop.c.

7.19 /usr/src/sys/netgraph/atm/sscop/ng_sscop_cust.h File Reference

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <machine/stdarg.h>
#include <netnatm/saal/sscopdef.h>
```

Include dependency graph for ng_sscop_cust.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define [MEMINIT\(\)](#)
- #define [MEMZALLOC\(PTR, CAST, SIZE\)](#) ((PTR) = (CAST)malloc((SIZE), M_NG_SSCOP, M_NOWAIT | M_ZERO))
- #define [MEMFREE\(PTR\)](#) free((PTR), M_NG_SSCOP)
- #define [MSG_ALLOC\(PTR\)](#) MEMZALLOC(PTR, struct sscop_msg *, sizeof(struct sscop_msg))
- #define [MSG_FREE\(PTR\)](#) MEMFREE(PTR)
- #define [SIG_ALLOC\(PTR\)](#) MEMZALLOC(PTR, struct sscop_sig *, sizeof(struct sscop_sig))
- #define [SIG_FREE\(PTR\)](#) MEMFREE(PTR)
- #define [TIMER_INIT\(S, T\)](#) ng_callout_init(&(S) → t_##T)
- #define [TIMER_STOP\(S, T\)](#)
- #define [TIMER_RESTART\(S, T\)](#)
- #define [TIMER_ISACT\(S, T\)](#) ((S) → t_##T.c_flags & (CALLOUT_PENDING))
- #define [TIMER_FUNC\(T, N\)](#)
- #define [MSGQ_EMPTY\(Q\)](#) TAILQ_EMPTY(Q)
- #define [MSGQ_INIT\(Q\)](#) TAILQ_INIT(Q)
- #define [MSGQ_FOREACH\(P, Q\)](#) TAILQ_FOREACH(P, Q, link)
- #define [MSGQ_REMOVE\(Q, M\)](#) TAILQ_REMOVE(Q, M, link)
- #define [MSGQ_INSERT_BEFORE\(B, M\)](#) TAILQ_INSERT_BEFORE(B, M, link)
- #define [MSGQ_APPEND\(Q, M\)](#) TAILQ_INSERT_TAIL(Q, M, link)
- #define [MSGQ_PEEK\(Q\)](#) TAILQ_FIRST((Q))
- #define [MSGQ_GET\(Q\)](#) ng_sscop_msgq_get((Q))
- #define [DECL_MSGQ_GET](#)
- #define [MSGQ_CLEAR\(Q\)](#)
- #define [SIGQ_INIT\(Q\)](#) TAILQ_INIT(Q)
- #define [SIGQ_APPEND\(Q, S\)](#) TAILQ_INSERT_TAIL(Q, S, link)
- #define [SIGQ_EMPTY\(Q\)](#) TAILQ_EMPTY(Q)
- #define [SIGQ_GET\(Q\)](#) ng_sscop_sigq_get((Q))
- #define [DECL_SIGQ_GET](#)
- #define [SIGQ_MOVE\(F, T\)](#)
- #define [SIGQ_PREPEND\(F, T\)](#)
- #define [SIGQ_CLEAR\(Q\)](#)
- #define [MBUF_FREE\(M\)](#) do { if ((M)) m_freem((M)); } while(0)
- #define [MBUF_DUP\(M\)](#) m_copypacket((M), M_NOWAIT)
- #define [MBUF_LEN\(M\)](#) ((size_t)(M) → m_pkthdr.len)
- #define [MBUF_TRAIL32\(M,I\)](#) ng_sscop_mbuf_trail32((M), (I))
- #define [MBUF_STRIP32\(M\)](#) ng_sscop_mbuf_strip32((M))
- #define [MBUF_GET32\(M\)](#) ng_sscop_mbuf_get32((M))
- #define [MBUF_APPEND32\(M, W\)](#)
- #define [MBUF_PAD4\(M\)](#) ng_sscop_mbuf_pad4((M))
- #define [MBUF_UNPAD\(M, P\)](#) do { if((P) > 0) m_adj((M), -(P)); } while (0)
- #define [MBUF_ALLOC\(N\)](#) ng_sscop_mbuf_alloc((N))
- #define [DECL_MBUF_ALLOC](#)
- #define [ASSERT\(X\)](#)

Typedefs

- typedef callout [sscop_timer_t](#)

Functions

- typedef [TAILQ_ENTRY](#) (sscop_msg) sscop_msgq_link_t
- typedef [TAILQ_HEAD](#) (sscop_msgq, sscop_msg) sscop_msgq_head_t
- typedef [TAILQ_ENTRY](#) (sscop_sig) sscop_sigq_link_t
- typedef [TAILQ_HEAD](#) (sscop_sigq, sscop_sig) sscop_sigq_head_t
- static uint32_t __inline [ng_sscop_mbuf_trail32](#) (const struct mbuf *m, int i)
- static uint32_t __inline [ng_sscop_mbuf_strip32](#) (struct mbuf *m)
- static uint32_t __inline [ng_sscop_mbuf_get32](#) (struct mbuf *m)
- static u_int __inline [ng_sscop_mbuf_pad4](#) (struct mbuf *m)

7.19.1 Define Documentation

7.19.1.1 #define ASSERT(X)

Definition at line 343 of file ng_sscop_cust.h.

7.19.1.2 #define DECL_MBUF_ALLOC

Value:

```
static __inline struct mbuf *
ng_sscop_mbuf_alloc(size_t n)
{
    struct mbuf *m;

    MGETHDR(m, M_NOWAIT, MT_DATA);
    if (m != NULL) {
        m->m_len = 0;
        m->m_pkthdr.len = 0;
        if (n > MHLEN) {
            MCLGET(m, M_NOWAIT);
            if (!(m->m_flags & M_EXT)){
                m_free(m);
                m = NULL;
            }
        }
    }
    return (m);
}
```

Definition at line 319 of file ng_sscop_cust.h.

7.19.1.3 #define DECL_MSGQ_GET

Value:

```
static __inline struct sscop_msg *
ng_sscop_msgq_get(struct sscop_msgq *q)
{
    struct sscop_msg *m;

    m = TAILQ_FIRST(q);
    if (m != NULL)
        TAILQ_REMOVE(q, m, link);
    return (m);
}
```

Definition at line 150 of file ng_sscop_cust.h.

7.19.1.4 #define DECL_SIGQ_GET

Value:

```
static __inline struct sscop_sig *
ng_sscop_sigq_get(struct sscop_sigq *q)
{
    struct sscop_sig *s;

    s = TAILQ_FIRST(q);
    if (s != NULL)
        TAILQ_REMOVE(q, s, link);
    return (s);
}
```

Definition at line 185 of file ng_sscop_cust.h.

7.19.1.5 #define MBUF_ALLOC(N) ng_sscop_mbuf_alloc((N))

Definition at line 317 of file ng_sscop_cust.h.

7.19.1.6 #define MBUF_APPEND32(M, W)

Value:

```
do {
    uint32_t _w = (W);

    _w = htonl(_w);
    m_copyback((M), (M)->m_pkthdr.len, 4, (caddr_t)&_w);
} while (0)
```

Definition at line 286 of file ng_sscop_cust.h.

7.19.1.7 #define MBUF_DUP(M) m_copypacket((M), M_NOWAIT)

Definition at line 237 of file ng_sscop_cust.h.

7.19.1.8 #define MBUF_FREE(M) do { if ((M)) m_freem((M)); } while(0)

Definition at line 236 of file ng_sscop_cust.h.

7.19.1.9 #define MBUF_GET32(M) ng_sscop_mbuf_get32((M))

Definition at line 271 of file ng_sscop_cust.h.

7.19.1.10 #define MBUF_LEN(M) ((size_t)(M) -> m_pkthdr.len)

Definition at line 238 of file ng_sscop_cust.h.

7.19.1.11 #define MBUF_PAD4(M) ng_sscop_mbuf_pad4((M))

Definition at line 298 of file ng_sscop_cust.h.

7.19.1.12 #define MBUF_STRIP32(M) ng_sscop_mbuf_strip32((M))

Definition at line 259 of file ng_sscop_cust.h.

7.19.1.13 #define MBUF_TRAIL32(M, I) ng_sscop_mbuf_trail32((M), (I))

Definition at line 245 of file ng_sscop_cust.h.

7.19.1.14 #define MBUF_UNPAD(M, P) do { if((P) > 0) m_adj((M), -(P)); } while (0)

Definition at line 312 of file ng_sscop_cust.h.

7.19.1.15 #define MEMFREE(PTR) free((PTR), M_NG_SSCOP)

Definition at line 63 of file ng_sscop_cust.h.

7.19.1.16 #define MEMINIT()

Value:

```
MALLOC_DECLARE(M_NG_SSCOP); \
    DECL_MSGQ_GET \
    DECL_SIGQ_GET \
    DECL_MBUF_ALLOC
```

Definition at line 55 of file ng_sscop_cust.h.

7.19.1.17 #define MEMZALLOC(PTR, CAST, SIZE) ((PTR) = (CAST)malloc((SIZE), M_NG_SSCOP, M_NOWAIT | M_ZERO))

Definition at line 61 of file ng_sscop_cust.h.

7.19.1.18 #define MSG_ALLOC(PTR) MEMZALLOC(PTR, struct sscop_msg *, sizeof(struct sscop_msg))

Definition at line 66 of file ng_sscop_cust.h.

7.19.1.19 #define MSG_FREE(PTR) MEMFREE(PTR)

Definition at line 68 of file ng_sscop_cust.h.

7.19.1.20 #define MSGQ_APPEND(Q, M) TAILQ_INSERT_TAIL(Q, M, link)

Definition at line 145 of file ng_sscop_cust.h.

7.19.1.21 #define MSGQ_CLEAR(Q)**Value:**

```
do {
    struct sscop_msg *_m1, *_m2;
    _m1 = TAILQ_FIRST(Q);
    while (_m1 != NULL) {
        _m2 = TAILQ_NEXT(_m1, link);
        SSCOP_MSG_FREE(_m1);
        _m1 = _m2;
    }
    TAILQ_INIT((Q));
} while (0)
```

Definition at line 162 of file ng_sscop_cust.h.

7.19.1.22 #define MSGQ_EMPTY(Q) TAILQ_EMPTY(Q)

Definition at line 140 of file ng_sscop_cust.h.

7.19.1.23 #define MSGQ_FOREACH(P, Q) TAILQ_FOREACH(P, Q, link)

Definition at line 142 of file ng_sscop_cust.h.

7.19.1.24 #define MSGQ_GET(Q) ng_sscop_msgq_get((Q))

Definition at line 148 of file ng_sscop_cust.h.

7.19.1.25 #define MSGQ_INIT(Q) TAILQ_INIT(Q)

Definition at line 141 of file ng_sscop_cust.h.

7.19.1.26 #define MSGQ_INSERT_BEFORE(B, M) TAILQ_INSERT_BEFORE(B, M, link)

Definition at line 144 of file ng_sscop_cust.h.

7.19.1.27 #define MSGQ_PEEK(Q) TAILQ_FIRST((Q))

Definition at line 146 of file ng_sscop_cust.h.

7.19.1.28 #define MSGQ_REMOVE(Q, M) TAILQ_REMOVE(Q, M, link)

Definition at line 143 of file ng_sscop_cust.h.

7.19.1.29 #define SIG_ALLOC(PTR) MEMZALLOC(PTR, struct sscop_sig *, sizeof(struct sscop_sig))

Definition at line 71 of file ng_sscop_cust.h.

7.19.1.30 #define SIG_FREE(PTR) MEMFREE(PTR)

Definition at line 73 of file ng_sscop_cust.h.

7.19.1.31 #define SIGQ_APPEND(Q, S) TAILQ_INSERT_TAIL(Q, S, link)

Definition at line 181 of file ng_sscop_cust.h.

7.19.1.32 #define SIGQ_CLEAR(Q)**Value:**

```
do {
    struct sscop_sig *_s1, *_s2;
    _s1 = TAILQ_FIRST(Q);
    while (_s1 != NULL) {
        _s2 = TAILQ_NEXT(_s1, link);
        SSCOP_MSG_FREE(_s1->msg);
        SIG_FREE(_s1);
        _s1 = _s2;
    }
    TAILQ_INIT(Q);
} while (0)
```

Definition at line 219 of file ng_sscop_cust.h.

7.19.1.33 #define SIGQ_EMPTY(Q) TAILQ_EMPTY(Q)

Definition at line 182 of file ng_sscop_cust.h.

7.19.1.34 #define SIGQ_GET(Q) ng_sscop_sigq_get((Q))

Definition at line 184 of file ng_sscop_cust.h.

7.19.1.35 #define SIGQ_INIT(Q) TAILQ_INIT(Q)

Definition at line 180 of file ng_sscop_cust.h.

7.19.1.36 #define SIGQ_MOVE(F, T)**Value:**

```
do {
    struct sscop_sig *_s;
    while (!TAILQ_EMPTY(F)) {
        _s = TAILQ_FIRST(F);
        TAILQ_REMOVE(F, _s, link);
        TAILQ_INSERT_TAIL(T, _s, link);
    }
} while (0)
```

Definition at line 197 of file ng_sscop_cust.h.

7.19.1.37 #define SIGQ_PREPEND(F, T)**Value:**

```
do {
    struct sscop_sig *_s;
    while (!TAILQ_EMPTY(F)) {
        _s = TAILQ_LAST(F, sscop_sigq);
        TAILQ_REMOVE(F, _s, link);
        TAILQ_INSERT_HEAD(T, _s, link);
    }
} while (0)
```

Definition at line 208 of file ng_sscop_cust.h.

7.19.1.38 #define TIMER_FUNC(T, N)**Value:**

```
static void
T##_func(node_p node, hook_p hook, void *arg1, int arg2)
{
    struct sscop *sscop = arg1;
    VERBOSE(sscop, SSCOP_DBG_TIMER, (sscop, sscop->aarg,
        "timer_" #T " expired"));
    sscop_signal(sscop, SIG_T_##N, NULL);
}
```

Definition at line 123 of file ng_sscop_cust.h.

7.19.1.39 #define TIMER_INIT(S, T) ng_callout_init(&(S) → t_##T)

Definition at line 109 of file ng_sscop_cust.h.

7.19.1.40 #define TIMER_ISACT(S, T) ((S) → t_##T.c_flags & (CALLOUT_PENDING))

Definition at line 118 of file ng_sscop_cust.h.

7.19.1.41 #define TIMER_RESTART(S, T)**Value:**

```
do {
    TIMER_STOP(S, T);
    ng_callout(&(S)->t_##T, (S)->aarg, NULL,
        hz * (S)->timer##T / 1000, T##_func, (S), 0);
} while (0)
```

Definition at line 113 of file ng_sscop_cust.h.

7.19.1.42 #define TIMER_STOP(S, T)**Value:**

```
do {
    ng_untimeout (&(S)->t_##T, (S)->aarg);
} while (0)
```

Definition at line 110 of file ng_sscop_cust.h.

7.19.2 Typedef Documentation**7.19.2.1 typedef struct callout [sscop_timer_t](#)**

Definition at line 108 of file ng_sscop_cust.h.

7.19.3 Function Documentation**7.19.3.1 static uint32_t __inline ng_sscop_mbuf_get32 (struct mbuf * *m*) [static]**

Definition at line 274 of file ng_sscop_cust.h.

7.19.3.2 static u_int __inline ng_sscop_mbuf_pad4 (struct mbuf * *m*) [static]

Definition at line 301 of file ng_sscop_cust.h.

7.19.3.3 static uint32_t __inline ng_sscop_mbuf_strip32 (struct mbuf * *m*) [static]

Definition at line 262 of file ng_sscop_cust.h.

7.19.3.4 static uint32_t __inline ng_sscop_mbuf_trail32 (const struct mbuf * *m*, int *i*) [static]

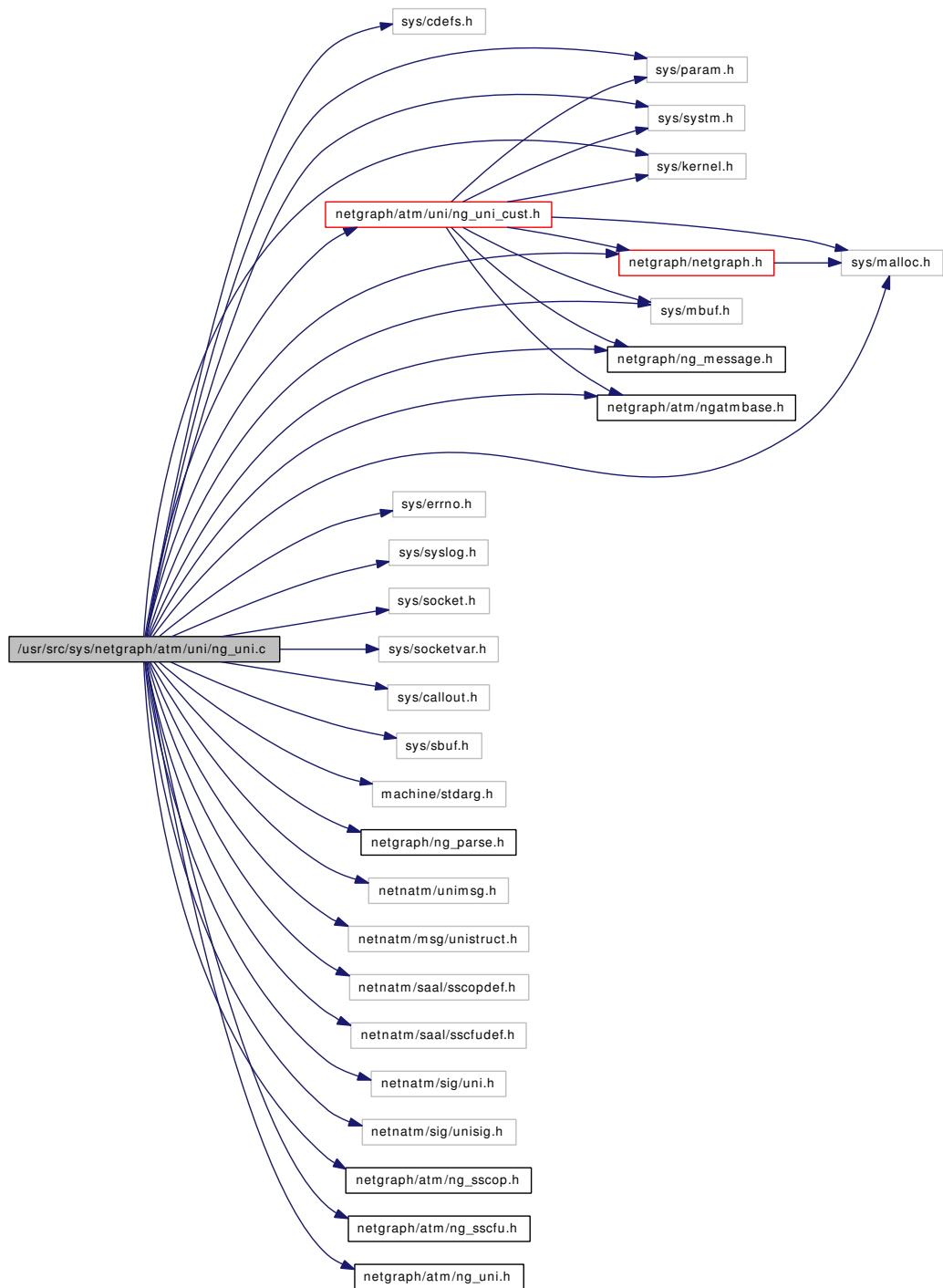
Definition at line 248 of file ng_sscop_cust.h.

7.19.3.5 typedef TAILQ_ENTRY (sscop_sig)**7.19.3.6 typedef TAILQ_ENTRY (sscop_msg)****7.19.3.7 typedef TAILQ_HEAD (sscop_sigq, sscop_sig)****7.19.3.8 typedef TAILQ_HEAD (sscop_msgq, sscop_msg)**

7.20 /usr/src/sys/netgraph/atm/uni/ng_uni.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/callout.h>
#include <sys/sbuf.h>
#include <machine/stdarg.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netnatm/unimsg.h>
#include <netnatm/msg/unistruct.h>
#include <netgraph/atm/ngatmbase.h>
#include <netnatm/saal/sscopdef.h>
#include <netnatm/saal/sscfudef.h>
#include <netgraph/atm/uni/ng_uni_cust.h>
#include <netnatm/sig/uni.h>
#include <netnatm/sig/unisig.h>
#include <netgraph/atm/ng_sscop.h>
#include <netgraph/atm/ng_sscfu.h>
#include <netgraph/atm/ng_uni.h>
```

Include dependency graph for ng_uni.c:



Data Structures

- struct [priv](#)
- struct [unimem_debug](#)

Defines

- #define **D(S)** case S: printf("%s", #S); break
- #define **UNI_DEBUG_DEFINE(D)** [UNI_FAC_##D] #D,

Functions

- **__FBSDID** ("FreeBSD: src/sys/netgraph/atm/uni/ng_uni.c,v 1.6 2005/10/31 15:41:26 rwatson Exp \$")
- **MALLOC_DEFINE** (M_NG_UNI,"netgraph_uni_node","netgraph uni node")
- **MALLOC_DEFINE** (M_UNI,"netgraph_uni_data","uni protocol data")
- **MODULE_DEPEND** (ng_uni, ngatmbase, 1, 1, 1)
- static int **ng_uni_mod_event** (module_t, int, void *)
- **NETGRAPH_INIT** (uni,&ng_uni_typestruct)
- static void **uni_uni_output** (struct uni *, void *, enum uni_sig, u_int32_t, struct uni_msg *)
- static void **uni_saal_output** (struct uni *, void *, enum saal_sig, struct uni_msg *)
- static void **uni_verbose** (struct uni *, static void uni_do_status(struct uni void *, u_int, static void uni_do_status(struct uni const char *,...))
- static int **ng_uni_constructor** (node_p node)
- static int **ng_uni_shutdown** (node_p node)
- static void **uni_do_status** (struct uni *uni, void *uarg, void *sbuf, const char *fmt,...)
- static int **text_status** (node_p node, struct priv *priv, char *buf, u_int len)
- static int **ng_uni_rcvmsg** (node_p node, item_p item, hook_p lasthook)
- static int **ng_uni_newhook** (node_p node, hook_p hook, const char *name)
- static int **ng_uni_disconnect** (hook_p hook)
- static int **ng_uni_rcvupper** (hook_p hook, item_p item)
- static void **dump_uni_msg** (struct uni_msg *msg)
- static void **dump_saal_signal** (node_p node, enum saal_sig sig, struct uni_msg *msg, int to)
- static int **ng_uni_rcvlower** (hook_p hook __unused, item_p item)
- static void **uni_verbose** (struct uni *uni, void *varg, u_int fac, const char *fmt,...)
- **LIST_HEAD** (unimem_debug_list, unimem_debug)
- static void **uni_init** (void)
- static void **uni_fini** (void)
- void * **ng_uni_malloc** (enum unimem type, const char *file, u_int lno)
- void **ng_uni_free** (enum unimem type, void *ptr, const char *file, u_int lno)

Variables

- static struct ng_parse_struct_field ng_uni_config_mask_type_info []
- static struct ng_parse_type ng_uni_config_mask_type
- static struct ng_parse_struct_field ng_uni_config_type_info []
- static struct ng_parse_type ng_uni_config_type
- static struct ng_parse_struct_field ng_uni_set_config_type_info []
- static struct ng_parse_type ng_uni_set_config_type
- static struct ng_parse_fixedarray_info ng_uni_debuglevel_type_info
- static struct ng_parse_type ng_uni_debuglevel_type
- static struct ng_parse_struct_field ng_uni_debug_type_info []
- static struct ng_parse_type ng_uni_debug_type
- static struct ng_cmdlist ng_uni_cmdlist []
- static ng_constructor_t ng_uni_constructor

- static `ng_shutdown_t ng_uni_shutdown`
- static `ng_rcvmsg_t ng_uni_rcvmsg`
- static `ng_newhook_t ng_uni_newhook`
- static `ng_disconnect_t ng_uni_disconnect`
- static `ng_rcvdata_t ng_uni_rcvlower`
- static `ng_rcvdata_t ng_uni_rcvupper`
- static struct `ng_type ng_uni_typestruct`
- static struct unimem_debug_list `nguni_freemem` [UNIMEM_TYPES]
- static struct unimem_debug_list `nguni_usedmem` [UNIMEM_TYPES]
- static struct mtx `nguni_unilist_mtx`
- static const char * `unimem_names` [UNIMEM_TYPES]

7.20.1 Define Documentation

7.20.1.1 #define D(S) case S: printf("%s", #S); break

Referenced by `dump_saal_signal()`.

7.20.1.2 #define UNI_DEBUG_DEFINE(D) [UNI_FAC_##D] #D,

7.20.2 Function Documentation

7.20.2.1 __FBSDID ("\$FreeBSD: src/sys/netgraph/atm/uni/ng_uni. c, v 1.6 2005/10/31 15:41:26 rwatson Exp \$")

7.20.2.2 static void dump_saal_signal (`node_p node`, enum `saal_sig sig`, struct `uni_msg * msg`, int `to`) [static]

Definition at line 620 of file `ng_uni.c`.

References `D`, `dump_uni_msg()`, `NG_NODE_PRIVATE`, and `priv::uni`.

Referenced by `ng_uni_rcvlower()`, and `uni_saal_output()`.

Here is the call graph for this function:



7.20.2.3 static void dump_uni_msg (struct `uni_msg * msg`) [static]

Definition at line 598 of file `ng_uni.c`.

Referenced by `dump_saal_signal()`.

7.20.2.4 LIST_HEAD (unimem_debug_list, unimem_debug)**7.20.2.5** MALLOC_DEFINE (M_UNI, "netgraph_uni_data", "uni protocol data")**7.20.2.6** MALLOC_DEFINE (M_NG_UNI, "netgraph_uni_node", "netgraph uni node")**7.20.2.7** MODULE_DEPEND (ng_uni, ngatmbase, 1, 1, 1)**7.20.2.8** NETGRAPH_INIT (uni, & ng_uni_tpestruct)**7.20.2.9** static int ng_uni_constructor (node_p node) [static]

Definition at line 222 of file ng_uni.c.

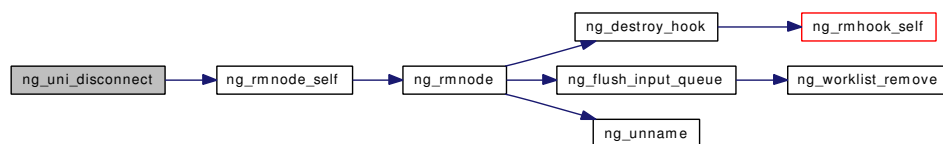
References NG_NODE_FORCE_WRITER, and NG_NODE_SET_PRIVATE.

7.20.2.10 static int ng_uni_disconnect (hook_p hook) [static]

Definition at line 492 of file ng_uni.c.

References priv::lower, NG_HOOK_NAME, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), and priv::upper.

Here is the call graph for this function:

**7.20.2.11** void ng_uni_free (enum unimem type, void * ptr, const char * file, u_int lno)

Definition at line 866 of file ng_uni.c.

References unimem_debug::file, unimem_debug::lno, nguni_freemem, and unimem_names.

7.20.2.12 void* ng_uni_malloc (enum unimem type, const char * file, u_int lno)

Definition at line 833 of file ng_uni.c.

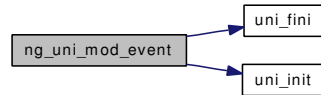
References nguni_freemem, and unimem_sizes.

7.20.2.13 static int ng_uni_mod_event (module_t, int, void *) [static]

Definition at line 909 of file ng_uni.c.

References uni_fini(), and uni_init().

Here is the call graph for this function:



7.20.2.14 `static int ng_uni_newhook (node_p node, hook_p hook, const char * name) [static]`

Definition at line 476 of file ng_uni.c.

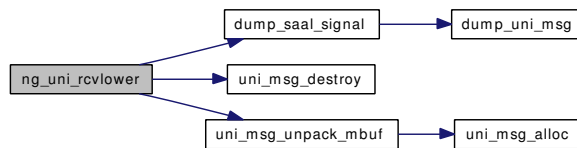
References `priv::lower`, `NG_HOOK_SET_RCVDATA`, `NG_NODE_PRIVATE`, `ng_uni_rcvupper`, and `priv::upper`.

7.20.2.15 `static int ng_uni_rcvlower (hook_p hook __unused, item_p item) [static]`

Definition at line 661 of file ng_uni.c.

References `dump_saal_signal()`, `priv::enabled`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `priv::uni`, `uni_msg_destroy()`, and `uni_msg_unpack_mbuf()`.

Here is the call graph for this function:



7.20.2.16 `static int ng_uni_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]`

Definition at line 306 of file ng_uni.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `priv::enabled`, `ng_mesg::header`, `ngm_uni_set_config::mask`, `ngm_uni_config_mask::mask`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_TEXTRESPONSE`, `NGI_GET_MSG`, `NGM_GENERIC_COOKIE`, `NGM_TEXT_STATUS`, `NGM_UNI_COOKIE`, `NGM_UNI_DISABLE`, `NGM_UNI_ENABLE`, `NGM_UNI_GET_CONFIG`, `NGM_UNI_GETDEBUG`, `NGM_UNI_GETSTATE`, `NGM_UNI_SET_CONFIG`, `NGM_UNI_SETDEBUG`, `text_status()`, `ng_mesg::ng_msghdr::typecookie`, and `priv::uni`.

Here is the call graph for this function:

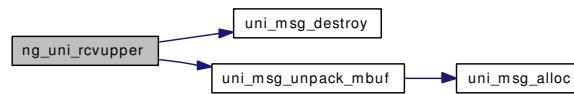


7.20.2.17 `static int ng_uni_rcvupper (hook_p hook, item_p item) [static]`

Definition at line 522 of file ng_uni.c.

References `priv::enabled`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `priv::uni`, `uni_msg_destroy()`, and `uni_msg_unpack_mbuf()`.

Here is the call graph for this function:



7.20.2.18 static int ng_uni_shutdown (node_p node) [static]

Definition at line 241 of file ng_uni.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and priv::uni.

7.20.2.19 static int text_status (node_p node, struct priv *priv, char *buf, u_int len) [static]

Definition at line 270 of file ng_uni.c.

References priv::lower, NG_HOOK_NAME, NG_HOOK_NODE, NG_HOOK_PEER, NG_NODE_NAME, priv::uni, and priv::upper.

7.20.2.20 static void uni_do_status (struct uni *uni, void *uarg, void *sbuf, const char *fmt, ...) [static]

Definition at line 260 of file ng_uni.c.

Referenced by uni_verbose().

7.20.2.21 static void uni_fini (void) [static]

Definition at line 806 of file ng_uni.c.

References nguni_freemem, and unimem_names.

Referenced by ng_uni_mod_event().

7.20.2.22 static void uni_init (void) [static]

Definition at line 799 of file ng_uni.c.

Referenced by ng_uni_mod_event().

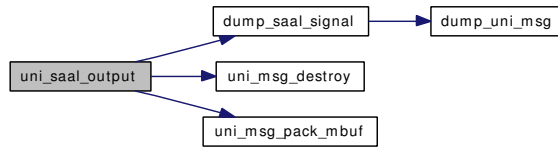
7.20.2.23 static void uni_saal_output (struct uni *, void *, enum saal_sig, struct uni_msg *) [static]

Definition at line 712 of file ng_uni.c.

References dump_saal_signal(), priv::lower, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, sscfu_arg::sig, priv::uni, uni_msg_destroy(), and uni_msg_pack_mbuf().

Referenced by uni_verbose().

Here is the call graph for this function:



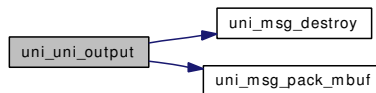
7.20.2.24 static void uni_uni_output (struct uni *, void *, enum uni_sig, u_int32_t, struct uni_msg *) [static]

Definition at line 570 of file ng_uni.c.

References uni_arg::cookie, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, uni_arg::sig, uni_msg_destroy(), uni_msg_pack_mbuf(), and priv::upper.

Referenced by uni_verbose().

Here is the call graph for this function:



7.20.2.25 static void uni_verbose (struct uni * uni, void * varg, u_int fac, const char * fmt, ...) [static]

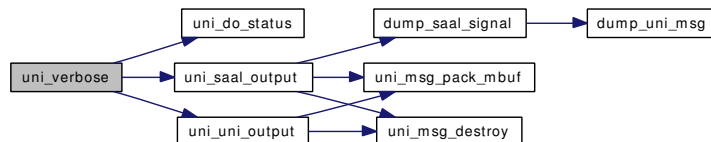
Definition at line 741 of file ng_uni.c.

7.20.2.26 static void uni_verbose (struct uni *, static void uni_do_status(struct uni void *, u_int, static void uni_do_status(struct uni const char *, ...) [static]

Definition at line 205 of file ng_uni.c.

References uni_do_status(), uni_saal_output(), and uni_uni_output().

Here is the call graph for this function:



7.20.3 Variable Documentation

7.20.3.1 struct ng_cmdlist ng_uni_cmdlist[] [static]

Definition at line 121 of file ng_uni.c.

7.20.3.2 `struct ng_parse_type ng_uni_config_mask_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_uni_config_mask_type_info
}
```

Definition at line 81 of file ng_uni.c.

7.20.3.3 `struct ng_parse_struct_field ng_uni_config_mask_type_info[]` [static]**Initial value:**

```
NGM_UNI_CONFIG_MASK_INFO
```

Definition at line 79 of file ng_uni.c.

7.20.3.4 `struct ng_parse_type ng_uni_config_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_uni_config_type_info
}
```

Definition at line 89 of file ng_uni.c.

7.20.3.5 `struct ng_parse_struct_field ng_uni_config_type_info[]` [static]**Initial value:**

```
NGM_UNI_CONFIG_INFO
```

Definition at line 87 of file ng_uni.c.

7.20.3.6 `ng_constructor_t ng_uni_constructor` [static]

Definition at line 177 of file ng_uni.c.

7.20.3.7 `struct ng_parse_type ng_uni_debug_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_uni_debug_type_info
}
```

Definition at line 113 of file ng_uni.c.

7.20.3.8 `struct ng_parse_struct_field ng_uni_debug_type_info[]` [static]

Initial value:

```
NGM_UNI_DEBUG_INFO
```

Definition at line 111 of file ng_uni.c.

7.20.3.9 `struct ng_parse_type ng_uni_debuglevel_type` [static]

Initial value:

```
{
    &ng_parse_fixedarray_type,
    &ng_uni_debuglevel_type_info
}
```

Definition at line 107 of file ng_uni.c.

7.20.3.10 `struct ng_parse_fixedarray_info ng_uni_debuglevel_type_info` [static]

Initial value:

```
NGM_UNI_DEBUGLEVEL_INFO
```

Definition at line 105 of file ng_uni.c.

7.20.3.11 `ng_disconnect_t ng_uni_disconnect` [static]

Definition at line 181 of file ng_uni.c.

7.20.3.12 `ng_newhook_t ng_uni_newhook` [static]

Definition at line 180 of file ng_uni.c.

7.20.3.13 `ng_rcvdata_t ng_uni_rcvlower` [static]

Definition at line 182 of file ng_uni.c.

7.20.3.14 `ng_rcvmsg_t ng_uni_rcvmsg` [static]

Definition at line 179 of file ng_uni.c.

7.20.3.15 `ng_rcvdata_t ng_uni_rcvupper` [static]

Definition at line 183 of file ng_uni.c.

Referenced by ng_uni_newhook().

7.20.3.16 struct `ng_parse_type ng_uni_set_config_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    ng_uni_set_config_type_info
}
```

Definition at line 97 of file `ng_uni.c`.

7.20.3.17 struct `ng_parse_struct_field ng_uni_set_config_type_info[]` [static]**Initial value:**

```
NGM_UNI_SET_CONFIG_INFO
```

Definition at line 95 of file `ng_uni.c`.

7.20.3.18 `ng_shutdown_t ng_uni_shutdown` [static]

Definition at line 178 of file `ng_uni.c`.

7.20.3.19 struct `ng_type ng_uni_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_UNI_NODE_TYPE,
    .mod_event =   ng_uni_mod_event,
    .constructor = ng_uni_constructor,
    .rcvmsg =      ng_uni_rcvmsg,
    .shutdown =    ng_uni_shutdown,
    .newhook =     ng_uni_newhook,
    .rcvdata =     ng_uni_rcvlower,
    .disconnect =  ng_uni_disconnect,
    .cmdlist =     ng_uni_cmdlist,
}
```

Definition at line 187 of file `ng_uni.c`.

7.20.3.20 struct `unimem_debug_list nguni_freemem[UNIMEM_TYPES]` [static]**Initial value:**

```
{
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
}
```

Definition at line 773 of file `ng_uni.c`.

Referenced by `ng_uni_free()`, `ng_uni_malloc()`, and `uni_fini()`.

7.20.3.21 struct mtx [nguni_unilist_mtx](#) [static]

Definition at line 788 of file ng_uni.c.

7.20.3.22 struct unimem_debug_list [nguni_usedmem](#)[UNIMEM_TYPES] [static]**Initial value:**

```
{
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
    LIST_HEAD_INITIALIZER(unimem_debug),
}
```

Definition at line 780 of file ng_uni.c.

7.20.3.23 const char* [unimem_names](#)[UNIMEM_TYPES] [static]**Initial value:**

```
{
    "instance",
    "all",
    "signal",
    "call",
    "party"
}
```

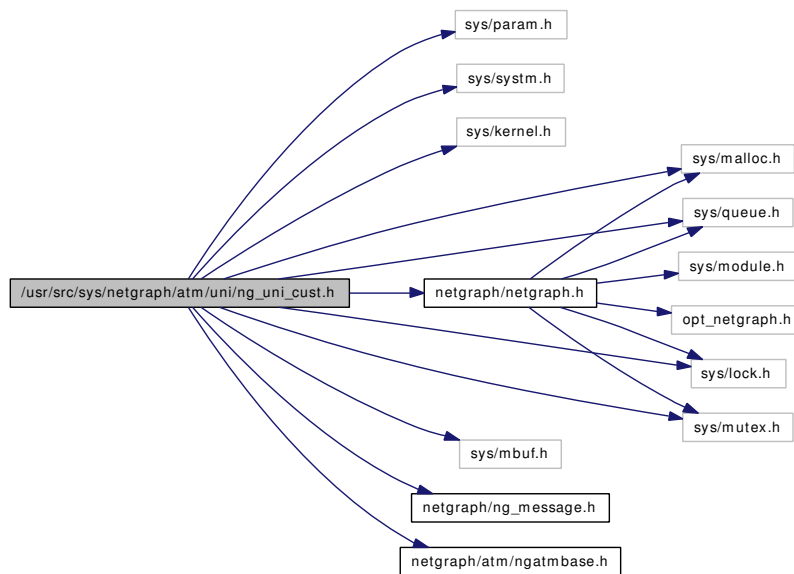
Definition at line 790 of file ng_uni.c.

Referenced by [ng_uni_free\(\)](#), and [uni_fini\(\)](#).

7.21 /usr/src/sys/netgraph/atm/uni/ng_uni_cust.h File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/queue.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/atm/ngatmbase.h>
```

Include dependency graph for ng_uni_cust.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [uni_timer](#)

Defines

- #define `ASSERT(E, M)` `KASSERT(E,M)`
- #define `UNIMEM_TYPES` `5`
- #define `INS_ALLOC()` `ng_uni_malloc(UNIMEM_INS, __FILE__, __LINE__)`
- #define `INS_FREE(P)` `ng_uni_free(UNIMEM_INS, P, __FILE__, __LINE__)`
- #define `UNI_ALLOC()` `ng_uni_malloc(UNIMEM_ALL, __FILE__, __LINE__)`
- #define `UNI_FREE(P)` `ng_uni_free(UNIMEM_ALL, P, __FILE__, __LINE__)`
- #define `SIG_ALLOC()` `ng_uni_malloc(UNIMEM_SIG, __FILE__, __LINE__)`
- #define `SIG_FREE(P)` `ng_uni_free(UNIMEM_SIG, P, __FILE__, __LINE__)`
- #define `CALL_ALLOC()` `ng_uni_malloc(UNIMEM_CALL, __FILE__, __LINE__)`
- #define `CALL_FREE(P)` `ng_uni_free(UNIMEM_CALL, P, __FILE__, __LINE__)`
- #define `PARTY_ALLOC()` `ng_uni_malloc(UNIMEM_PARTY, __FILE__, __LINE__)`
- #define `PARTY_FREE(P)` `ng_uni_free(UNIMEM_PARTY, P, __FILE__, __LINE__)`
- #define `_TIMER_INIT(X, T)` `ng_callout_init(&(X) → T.c)`
- #define `_TIMER_DESTROY(UNI, FIELD)` `_TIMER_STOP(UNI,FIELD)`
- #define `_TIMER_STOP(UNI, FIELD)`
- #define `TIMER_ISACT(UNI, T)`
- #define `_TIMER_START(UNI, ARG, FIELD, DUE, FUNC)`
- #define `TIMER_FUNC_UNI(T, F)`
- #define `TIMER_FUNC_CALL(T, F)`
- #define `TIMER_FUNC_PARTY(T, F)`
- #define `UNICORE`
- #define `memmove(T, F, L)` `bcopy((F), (T), (L))`

Enumerations

- enum `unimem` {
`UNIMEM_INS = 0, UNIMEM_ALL, UNIMEM_SIG, UNIMEM_CALL,`
`UNIMEM_PARTY }`

Functions

- void * `ng_uni_malloc` (enum `unimem`, const char *, u_int)
- void `ng_uni_free` (enum `unimem`, void *, const char *, u_int)

Variables

- size_t `unimem_sizes` [`UNIMEM_TYPES`]

7.21.1 Define Documentation

7.21.1.1 #define `_TIMER_DESTROY(UNI, FIELD)` `_TIMER_STOP(UNI,FIELD)`

Definition at line 86 of file `ng_uni_cust.h`.

7.21.1.2 #define `_TIMER_INIT(X, T)` `ng_callout_init(&(X) → T.c)`

Definition at line 85 of file `ng_uni_cust.h`.

7.21.1.3 #define _TIMER_START(UNI, ARG, FIELD, DUE, FUNC)**Value:**

```
do {
    \
    _TIMER_STOP(UNI, FIELD);
    ng_callout(&FIELD.c, (UNI)->arg, NULL,
              hz * (DUE) / 1000, FUNC, (ARG), 0);
} while (0)
```

Definition at line 92 of file ng_uni_cust.h.

7.21.1.4 #define _TIMER_STOP(UNI, FIELD)**Value:**

```
do {
    ng_uncallout(&FIELD.c, (UNI)->arg);
} while (0)
```

Definition at line 87 of file ng_uni_cust.h.

7.21.1.5 #define ASSERT(E, M) KASSERT(E,M)

Definition at line 46 of file ng_uni_cust.h.

7.21.1.6 #define CALL_ALLOC() ng_uni_malloc(UNIMEM_CALL, __FILE__, __LINE__)

Definition at line 72 of file ng_uni_cust.h.

7.21.1.7 #define CALL_FREE(P) ng_uni_free(UNIMEM_CALL, P, __FILE__, __LINE__)

Definition at line 73 of file ng_uni_cust.h.

7.21.1.8 #define INS_ALLOC() ng_uni_malloc(UNIMEM_INS, __FILE__, __LINE__)

Definition at line 63 of file ng_uni_cust.h.

7.21.1.9 #define INS_FREE(P) ng_uni_free(UNIMEM_INS, P, __FILE__, __LINE__)

Definition at line 64 of file ng_uni_cust.h.

7.21.1.10 #define memmove(T, F, L) bcopy((F), (T), (L))

Definition at line 150 of file ng_uni_cust.h.

Referenced by ng_l2tp_seq_recv_nr().

7.21.1.11 #define PARTY_ALLOC() ng_uni_malloc(UNIMEM_PARTY, __FILE__, __LINE__)

Definition at line 75 of file ng_uni_cust.h.

7.21.1.12 #define PARTY_FREE(P) ng_uni_free(UNIMEM_PARTY, P, __FILE__, __LINE__)

Definition at line 76 of file ng_uni_cust.h.

7.21.1.13 #define SIG_ALLOC() ng_uni_malloc(UNIMEM_SIG, __FILE__, __LINE__)

Definition at line 69 of file ng_uni_cust.h.

7.21.1.14 #define SIG_FREE(P) ng_uni_free(UNIMEM_SIG, P, __FILE__, __LINE__)

Definition at line 70 of file ng_uni_cust.h.

7.21.1.15 #define TIMER_FUNC_CALL(T, F)**Value:**

```
static void F(struct call *);           \
static void                             \
_##T##_func(node_p node, hook_p hook, void *arg1, int arg2) \
{                                       \
    struct call *call = (struct call *)arg1; \
    struct uni *uni = call->uni;           \
                                           \
    (F)(call);                             \
    uni_work(uni);                         \
}
```

Definition at line 112 of file ng_uni_cust.h.

7.21.1.16 #define TIMER_FUNC_PARTY(T, F)**Value:**

```
static void F(struct party *);         \
static void                             \
_##T##_func(node_p node, hook_p hook, void *arg1, int arg2) \
{                                       \
    struct party *party = (struct party *)arg1; \
    struct uni *uni = party->call->uni; \
                                           \
    (F)(party);                             \
    uni_work(uni);                         \
}
```

Definition at line 127 of file ng_uni_cust.h.

7.21.1.17 #define TIMER_FUNC_UNI(T, F)**Value:**

```

static void F(struct uni *);
static void
_##T##_func(node_p node, hook_p hook, void *arg1, int arg2)
{
    struct uni *uni = (struct uni *)arg1;

    (F) (uni);
    uni_work(uni);
}

```

Definition at line 98 of file ng_uni_cust.h.

7.21.1.18 #define TIMER_ISACT(UNI, T)**Value:**

```

((UNI)->T.c.c_flags & (CALLOUT_ACTIVE | \
                                CALLOUT_PENDING))

```

Definition at line 90 of file ng_uni_cust.h.

7.21.1.19 #define UNI_ALLOC() ng_uni_malloc(UNIMEM_ALL, __FILE__, __LINE__)

Definition at line 66 of file ng_uni_cust.h.

7.21.1.20 #define UNI_FREE(P) ng_uni_free(UNIMEM_ALL, P, __FILE__, __LINE__)

Definition at line 67 of file ng_uni_cust.h.

7.21.1.21 #define UNICORE**Value:**

```

size_t unimem_sizes[UNIMEM_TYPES] = {
    [UNIMEM_INS]      = sizeof(struct uni),
    [UNIMEM_ALL]     = sizeof(struct uni_all),
    [UNIMEM_SIG]     = sizeof(struct sig),
    [UNIMEM_CALL]    = sizeof(struct call),
    [UNIMEM_PARTY]   = sizeof(struct party)
};

```

Definition at line 141 of file ng_uni_cust.h.

7.21.1.22 #define UNIMEM_TYPES 5

Definition at line 58 of file ng_uni_cust.h.

7.21.2 Enumeration Type Documentation

7.21.2.1 enum `unimem`

Enumerator:

`UNIMEM_INS`
`UNIMEM_ALL`
`UNIMEM_SIG`
`UNIMEM_CALL`
`UNIMEM_PARTY`

Definition at line 51 of file `ng_uni_cust.h`.

7.21.3 Function Documentation

7.21.3.1 void `ng_uni_free` (enum *unimem*, void *, const char *, u_int)

Definition at line 866 of file `ng_uni.c`.

References `unimem_debug::file`, `unimem_debug::lno`, `nguni_freemem`, and `unimem_names`.

7.21.3.2 void* `ng_uni_malloc` (enum *unimem*, const char *, u_int)

Definition at line 833 of file `ng_uni.c`.

References `nguni_freemem`, and `unimem_sizes`.

7.21.4 Variable Documentation

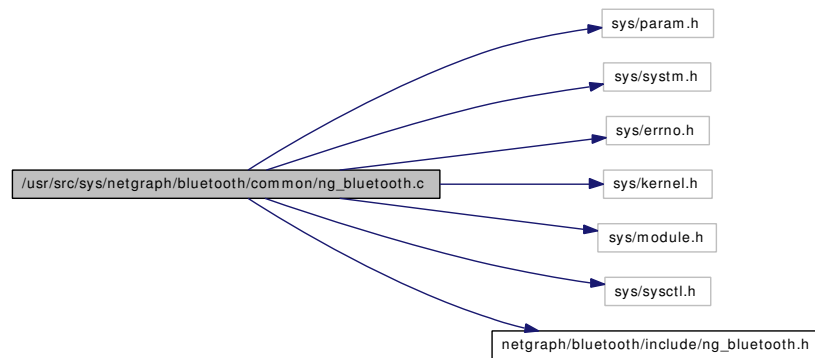
7.21.4.1 size_t `unimem_sizes`[UNIMEM_TYPES]

Referenced by `ng_uni_malloc()`.

7.22 /usr/src/sys/netgraph/bluetooth/common/ng_bluetooth.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/sysctl.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
```

Include dependency graph for ng_bluetooth.c:



Functions

- [SYSCTL_NODE](#) (`_net`, `OID_AUTO`, `bluetooth`, `CTLFLAG_RW`, 0, "Bluetooth family")
- [SYSCTL_INT](#) (`_net_bluetooth`, `OID_AUTO`, `version`, `CTLFLAG_RD`, 0, `NG_BLUETOOTH_VERSION`, "")
- [SYSCTL_NODE](#) (`_net_bluetooth`, `OID_AUTO`, `hci`, `CTLFLAG_RW`, 0, "Bluetooth HCI family")
- static int [bluetooth_set_hci_command_timeout_value](#) (`SYSCTL_HANDLER_ARGS`)
- [SYSCTL_PROC](#) (`_net_bluetooth_hci`, `OID_AUTO`, `command_timeout`, `CTLTYPE_INT|CTLFLAG_RW`, `&bluetooth_hci_command_timeout_value`, 5, `bluetooth_set_hci_command_timeout_value`, "I", "HCI command timeout (sec)")
- static int [bluetooth_set_hci_connect_timeout_value](#) (`SYSCTL_HANDLER_ARGS`)
- [SYSCTL_PROC](#) (`_net_bluetooth_hci`, `OID_AUTO`, `connection_timeout`, `CTLTYPE_INT|CTLFLAG_RW`, `&bluetooth_hci_connect_timeout_value`, 60, `bluetooth_set_hci_connect_timeout_value`, "I", "HCI connect timeout (sec)")
- [SYSCTL_INT](#) (`_net_bluetooth_hci`, `OID_AUTO`, `max_neighbor_age`, `CTLFLAG_RW`, `&bluetooth_hci_max_neighbor_age_value`, 600, "Maximal HCI neighbor cache entry age (sec)")
- [SYSCTL_NODE](#) (`_net_bluetooth`, `OID_AUTO`, `l2cap`, `CTLFLAG_RW`, 0, "Bluetooth L2CAP family")
- static int [bluetooth_set_l2cap_rtx_timeout_value](#) (`SYSCTL_HANDLER_ARGS`)
- [SYSCTL_PROC](#) (`_net_bluetooth_l2cap`, `OID_AUTO`, `rtx_timeout`, `CTLTYPE_INT|CTLFLAG_RW`, `&bluetooth_l2cap_rtx_timeout_value`, 60, `bluetooth_set_l2cap_rtx_timeout_value`, "I", "L2CAP RTX timeout (sec)")

- static int `bluetooth_set_l2cap_ertx_timeout_value` (SYSCTL_HANDLER_ARGS)
- `SYSCTL_PROC` (`_net_bluetooth_l2cap`, `OID_AUTO`, `ertx_timeout`, `CTLTYPE_INT|CTLFLAG_-RW`, `&bluetooth_l2cap_ertx_timeout_value`, `300`, `bluetooth_set_l2cap_ertx_timeout_value`, `"I"`, `"L2CAP ERTX timeout (sec)"`)
- `u_int32_t` `bluetooth_hci_command_timeout` (void)
- `u_int32_t` `bluetooth_hci_connect_timeout` (void)
- `u_int32_t` `bluetooth_hci_max_neighbor_age` (void)
- `u_int32_t` `bluetooth_l2cap_rtx_timeout` (void)
- `u_int32_t` `bluetooth_l2cap_ertx_timeout` (void)
- `SYSCTL_NODE` (`_net_bluetooth`, `OID_AUTO`, `rfcomm`, `CTLFLAG_RW`, `0`, `"Bluetooth RFCOMM family"`)
- static int `bluetooth_modevent` (module_t mod, int event, void *data)
- `DECLARE_MODULE` (`ng_bluetooth`, `bluetooth_mod`, `SI_SUB_PSEUDO`, `SI_ORDER_ANY`)
- `MODULE_VERSION` (`ng_bluetooth`, `NG_BLUETOOTH_VERSION`)

Variables

- static `u_int32_t` `bluetooth_hci_command_timeout_value` = 5
- static `u_int32_t` `bluetooth_hci_connect_timeout_value` = 60
- static `u_int32_t` `bluetooth_hci_max_neighbor_age_value` = 600
- static `u_int32_t` `bluetooth_l2cap_rtx_timeout_value` = 60
- static `u_int32_t` `bluetooth_l2cap_ertx_timeout_value` = 300
- static `moduledata_t` `bluetooth_mod`

7.22.1 Function Documentation

7.22.1.1 `u_int32_t` `bluetooth_hci_command_timeout` (void)

Definition at line 181 of file `ng_bluetooth.c`.

References `bluetooth_hci_command_timeout_value`.

Referenced by `ng_hci_command_timeout()`.

7.22.1.2 `u_int32_t` `bluetooth_hci_connect_timeout` (void)

Definition at line 187 of file `ng_bluetooth.c`.

References `bluetooth_hci_connect_timeout_value`.

Referenced by `ng_hci_con_timeout()`, and `ng_l2cap_lp_timeout()`.

7.22.1.3 `u_int32_t` `bluetooth_hci_max_neighbor_age` (void)

Definition at line 193 of file `ng_bluetooth.c`.

References `bluetooth_hci_max_neighbor_age_value`.

Referenced by `ng_hci_neighbor_stale()`.

7.22.1.4 u_int32_t bluetooth_l2cap_ertx_timeout (void)

Definition at line 205 of file ng_bluetooth.c.

References bluetooth_l2cap_ertx_timeout_value.

Referenced by ng_btsocket_l2cap_timeout(), and ng_l2cap_process_con_rsp().

7.22.1.5 u_int32_t bluetooth_l2cap_rtx_timeout (void)

Definition at line 199 of file ng_bluetooth.c.

References bluetooth_l2cap_rtx_timeout_value.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_con_wakeup().

7.22.1.6 static int bluetooth_modevent (module_t mod, int event, void * data) [static]

Definition at line 222 of file ng_bluetooth.c.

7.22.1.7 static int bluetooth_set_hci_command_timeout_value (SYSCTL_HANDLER_ARGS) [static]

Definition at line 69 of file ng_bluetooth.c.

References bluetooth_hci_command_timeout_value.

7.22.1.8 static int bluetooth_set_hci_connect_timeout_value (SYSCTL_HANDLER_ARGS) [static]

Definition at line 93 of file ng_bluetooth.c.

References bluetooth_hci_connect_timeout_value, and bluetooth_l2cap_rtx_timeout_value.

7.22.1.9 static int bluetooth_set_l2cap_ertx_timeout_value (SYSCTL_HANDLER_ARGS) [static]

Definition at line 153 of file ng_bluetooth.c.

References bluetooth_l2cap_ertx_timeout_value, and bluetooth_l2cap_rtx_timeout_value.

7.22.1.10 static int bluetooth_set_l2cap_rtx_timeout_value (SYSCTL_HANDLER_ARGS) [static]

Definition at line 128 of file ng_bluetooth.c.

References bluetooth_hci_connect_timeout_value, bluetooth_l2cap_ertx_timeout_value, and bluetooth_l2cap_rtx_timeout_value.

- 7.22.1.11 `DECLARE_MODULE` (`ng_bluetooth`, `bluetooth_mod`, `SI_SUB_PSEUDO`, `SI_ORDER_ANY`)
- 7.22.1.12 `MODULE_VERSION` (`ng_bluetooth`, `NG_BLUETOOTH_VERSION`)
- 7.22.1.13 `SYSCTL_INT` (`_net_bluetooth_hci`, `OID_AUTO`, `max_neighbor_age`, `CTLFLAG_RW`, `& bluetooth_hci_max_neighbor_age_value`, `600`, "Maximal HCI neighbor cache entry age (sec)")
- 7.22.1.14 `SYSCTL_INT` (`_net_bluetooth`, `OID_AUTO`, `version`, `CTLFLAG_RD`, `0`, `NG_BLUETOOTH_VERSION`, "")
- 7.22.1.15 `SYSCTL_NODE` (`_net_bluetooth`, `OID_AUTO`, `rfcomm`, `CTLFLAG_RW`, `0`, "Bluetooth RFCOMM family")
- 7.22.1.16 `SYSCTL_NODE` (`_net_bluetooth`, `OID_AUTO`, `l2cap`, `CTLFLAG_RW`, `0`, "Bluetooth L2CAP family")
- 7.22.1.17 `SYSCTL_NODE` (`_net_bluetooth`, `OID_AUTO`, `hci`, `CTLFLAG_RW`, `0`, "Bluetooth HCI family")
- 7.22.1.18 `SYSCTL_NODE` (`_net`, `OID_AUTO`, `bluetooth`, `CTLFLAG_RW`, `0`, "Bluetooth family")
- 7.22.1.19 `SYSCTL_PROC` (`_net_bluetooth_l2cap`, `OID_AUTO`, `ertx_timeout`, `CTLTYPE_INT` | `CTLFLAG_RW`, `& bluetooth_l2cap_ertx_timeout_value`, `300`, `bluetooth_set_l2cap_ertx_timeout_value`, "I", "L2CAP ERTX timeout (sec)")
- 7.22.1.20 `SYSCTL_PROC` (`_net_bluetooth_l2cap`, `OID_AUTO`, `rtx_timeout`, `CTLTYPE_INT` | `CTLFLAG_RW`, `& bluetooth_l2cap_rtx_timeout_value`, `60`, `bluetooth_set_l2cap_rtx_timeout_value`, "I", "L2CAP RTX timeout (sec)")
- 7.22.1.21 `SYSCTL_PROC` (`_net_bluetooth_hci`, `OID_AUTO`, `connection_timeout`, `CTLTYPE_INT` | `CTLFLAG_RW`, `& bluetooth_hci_connect_timeout_value`, `60`, `bluetooth_set_hci_connect_timeout_value`, "I", "HCI connect timeout (sec)")
- 7.22.1.22 `SYSCTL_PROC` (`_net_bluetooth_hci`, `OID_AUTO`, `command_timeout`, `CTLTYPE_INT` | `CTLFLAG_RW`, `& bluetooth_hci_command_timeout_value`, `5`, `bluetooth_set_hci_command_timeout_value`, "I", "HCI command timeout (sec)")

7.22.2 Variable Documentation

- 7.22.2.1 `u_int32_t bluetooth_hci_command_timeout_value = 5` [static]

Definition at line 47 of file `ng_bluetooth.c`.

Referenced by `bluetooth_hci_command_timeout()`, and `bluetooth_set_hci_command_timeout_value()`.

- 7.22.2.2 `u_int32_t bluetooth_hci_connect_timeout_value = 60` [static]

Definition at line 48 of file `ng_bluetooth.c`.

Referenced by `bluetooth_hci_connect_timeout()`, `bluetooth_set_hci_connect_timeout_value()`, and `bluetooth_set_l2cap_rtx_timeout_value()`.

7.22.2.3 u_int32_t bluetooth_hci_max_neighbor_age_value = 600 [static]

Definition at line 49 of file ng_bluetooth.c.

Referenced by bluetooth_hci_max_neighbor_age().

7.22.2.4 u_int32_t bluetooth_l2cap_ertx_timeout_value = 300 [static]

Definition at line 51 of file ng_bluetooth.c.

Referenced by bluetooth_l2cap_ertx_timeout(), bluetooth_set_l2cap_ertx_timeout_value(), and bluetooth_set_l2cap_rtx_timeout_value().

7.22.2.5 u_int32_t bluetooth_l2cap_rtx_timeout_value = 60 [static]

Definition at line 50 of file ng_bluetooth.c.

Referenced by bluetooth_l2cap_rtx_timeout(), bluetooth_set_hci_connect_timeout_value(), bluetooth_set_l2cap_ertx_timeout_value(), and bluetooth_set_l2cap_rtx_timeout_value().

7.22.2.6 moduledata_t bluetooth_mod [static]**Initial value:**

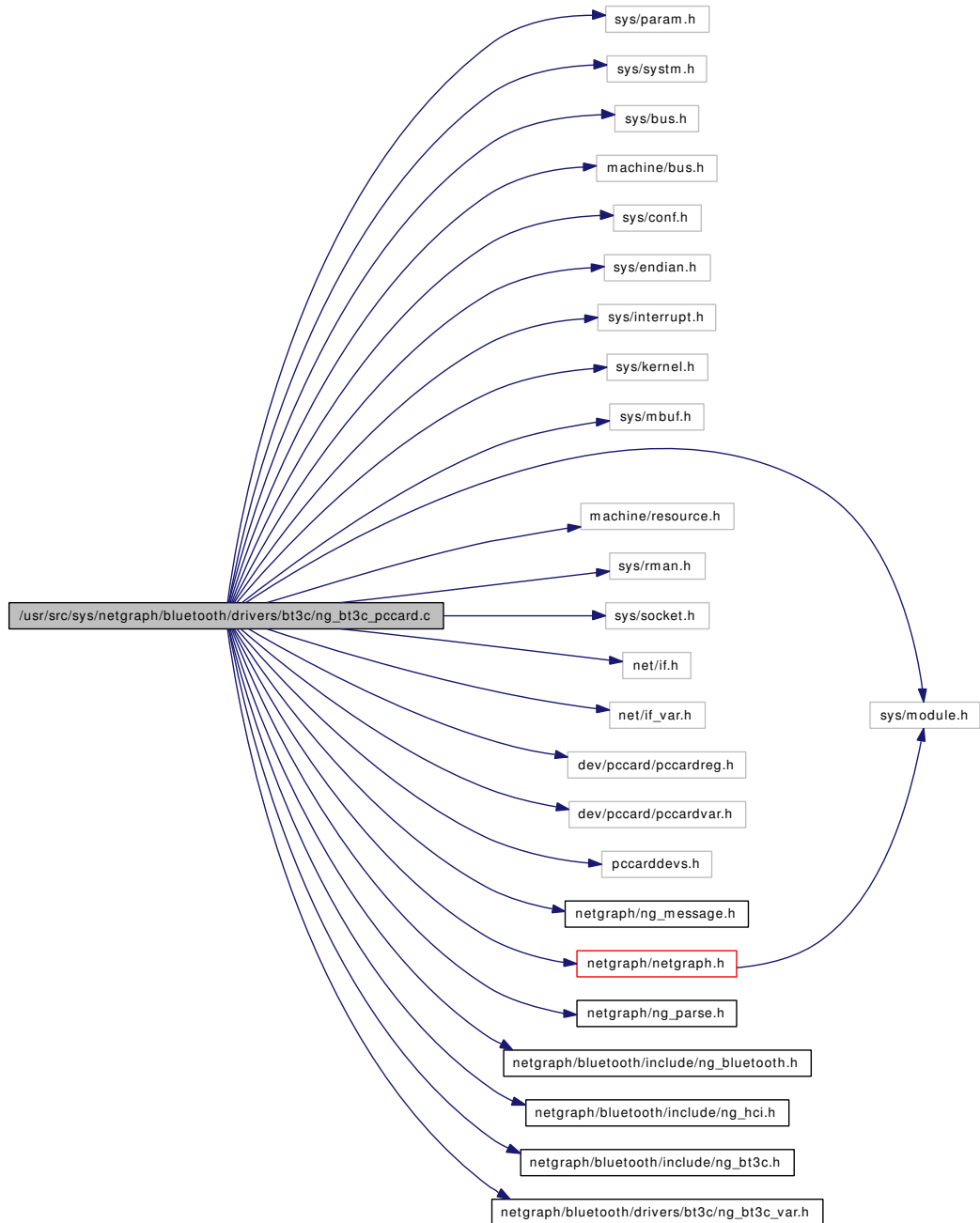
```
{
    "ng_bluetooth",
    bluetooth_modevent,
    NULL
}
```

Definition at line 245 of file ng_bluetooth.c.

7.23 /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_pccard.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bus.h>
#include <machine/bus.h>
#include <sys/conf.h>
#include <sys/endian.h>
#include <sys/interrupt.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/module.h>
#include <machine/resource.h>
#include <sys/rman.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/if_var.h>
#include <dev/pccard/pccardreg.h>
#include <dev/pccard/pccardvar.h>
#include "pccarddevs.h"
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_bt3c.h>
#include <netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h>
```

Include dependency graph for ng_bt3c_pccard.c:



Defines

- #define `bt3c_set_address`(sc, address)
- #define `bt3c_read_data`(sc, data)
- #define `bt3c_write_data`(sc, data)
- #define `bt3c_read_control`(sc, data)
- #define `bt3c_write_control`(sc, data)
- #define `bt3c_read`(sc, address, data)
- #define `bt3c_write`(sc, address, data)

Functions

- static int `bt3c_pccard_probe` (`device_t`)
- static int `bt3c_pccard_attach` (`device_t`)
- static int `bt3c_pccard_detach` (`device_t`)
- static void `bt3c_intr` (`void *`)
- static void `bt3c_receive` (`bt3c_softc_p`)
- static void `bt3c_swi_intr` (`void *`)
- static void `bt3c_forward` (`node_p`, `hook_p`, `void *`, `int`)
- static void `bt3c_send` (`node_p`, `hook_p`, `void *`, `int`)
- static void `bt3c_download_firmware` (`bt3c_softc_p`, `char const *`, `int`)
- static `MALLOC_DEFINE` (`M_BT3C`, "bt3c", "bt3c data structures")
- static int `ng_bt3c_constructor` (`node_p` `node`)
- static int `ng_bt3c_shutdown` (`node_p` `node`)
- static int `ng_bt3c_newhook` (`node_p` `node`, `hook_p` `hook`, `char const *name`)
- static int `ng_bt3c_connect` (`hook_p` `hook`)
- static int `ng_bt3c_disconnect` (`hook_p` `hook`)
- static int `ng_bt3c_rcvmsg` (`node_p` `node`, `item_p` `item`, `hook_p` `lasthook`)
- static int `ng_bt3c_rcvdata` (`hook_p` `hook`, `item_p` `item`)
- static int `bt3c_modevent` (`module_t` `mod`, `int` `event`, `void *data`)
- `DRIVER_MODULE` (`bt3c`, `pccard`, `bt3c_pccard_driver`, `bt3c_devclass`, `bt3c_modevent`, 0)
- `MODULE_VERSION` (`ng_bt3c`, `NG_BLUETOOTH_VERSION`)
- `MODULE_DEPEND` (`ng_bt3c`, `netgraph`, `NG_ABI_VERSION`, `NG_ABI_VERSION`, `NG_ABI_VERSION`)

Variables

- static `ng_constructor_t` `ng_bt3c_constructor`
- static `ng_shutdown_t` `ng_bt3c_shutdown`
- static `ng_newhook_t` `ng_bt3c_newhook`
- static `ng_connect_t` `ng_bt3c_connect`
- static `ng_disconnect_t` `ng_bt3c_disconnect`
- static `ng_rcvmsg_t` `ng_bt3c_rcvmsg`
- static `ng_rcvdata_t` `ng_bt3c_rcvdata`
- static struct `ng_parse_struct_field` `ng_bt3c_node_qlen_type_fields` []
- static struct `ng_parse_type` `ng_bt3c_node_qlen_type`
- static struct `ng_parse_struct_field` `ng_bt3c_node_stat_type_fields` []
- static struct `ng_parse_type` `ng_bt3c_node_stat_type`
- static struct `ng_cmdlist` `ng_bt3c_cmdlist` []
- static struct `ng_type` `typestruct`
- static `device_method_t` `bt3c_pccard_methods` []
- static `driver_t` `bt3c_pccard_driver`
- static `devclass_t` `bt3c_devclass`

7.23.1 Define Documentation

7.23.1.1 #define bt3c_read(sc, address, data)

Value:

```
do { \
    bt3c_set_address((sc), (address)); \
    bt3c_read_data((sc), (data)); \
} while(0)
```

Definition at line 125 of file ng_bt3c_pccard.c.

Referenced by bt3c_intr(), bt3c_receive(), and bt3c_swi_intr().

7.23.1.2 #define bt3c_read_control(sc, data)

Value:

```
do { \
    (data) = bus_space_read_1((sc)->iot, (sc)->ioh, BT3C_CONTROL); \
} while (0)
```

Definition at line 115 of file ng_bt3c_pccard.c.

Referenced by bt3c_download_firmware(), and bt3c_intr().

7.23.1.3 #define bt3c_read_data(sc, data)

Value:

```
do { \
    (data) = bus_space_read_1((sc)->iot, (sc)->ioh, BT3C_DATA_L); \
    (data) |= ((bus_space_read_1((sc)->iot, (sc)->ioh, BT3C_DATA_H) & 0xff) << 8); \
} while (0)
```

Definition at line 103 of file ng_bt3c_pccard.c.

7.23.1.4 #define bt3c_set_address(sc, address)

Value:

```
do { \
    bus_space_write_1((sc)->iot, (sc)->ioh, BT3C_ADDR_L, ((address) & 0xff)); \
    bus_space_write_1((sc)->iot, (sc)->ioh, BT3C_ADDR_H, (((address) >> 8) & 0xff)); \
} while (0)
```

Definition at line 97 of file ng_bt3c_pccard.c.

Referenced by bt3c_download_firmware(), bt3c_receive(), and bt3c_send().

7.23.1.5 #define bt3c_write(sc, address, data)**Value:**

```
do { \
    bt3c_set_address((sc), (address)); \
    bt3c_write_data((sc), (data)); \
} while (0)
```

Definition at line 131 of file ng_bt3c_pccard.c.

Referenced by bt3c_download_firmware(), bt3c_intr(), and bt3c_send().

7.23.1.6 #define bt3c_write_control(sc, data)**Value:**

```
do { \
    bus_space_write_1((sc)->iot, (sc)->ioh, BT3C_CONTROL, (data)); \
} while (0)
```

Definition at line 120 of file ng_bt3c_pccard.c.

Referenced by bt3c_download_firmware(), and bt3c_intr().

7.23.1.7 #define bt3c_write_data(sc, data)**Value:**

```
do { \
    bus_space_write_1((sc)->iot, (sc)->ioh, BT3C_DATA_L, ((data) & 0xff)); \
    bus_space_write_1((sc)->iot, (sc)->ioh, BT3C_DATA_H, (((data) >> 8) & 0xff)); \
} while (0)
```

Definition at line 109 of file ng_bt3c_pccard.c.

Referenced by bt3c_download_firmware(), and bt3c_send().

7.23.2 Function Documentation**7.23.2.1 static void bt3c_download_firmware (bt3c_softc_p, char const *, int) [static]**

Definition at line 1113 of file ng_bt3c_pccard.c.

References ng_bt3c_firmware_block_ep::block_address, ng_bt3c_firmware_block_ep::block_alignment, ng_bt3c_firmware_block_ep::block_size, bt3c_read_control, bt3c_set_address, bt3c_write, bt3c_write_control, bt3c_write_data, and bt3c_softc::dev.

Referenced by ng_bt3c_rcvmsg().

7.23.2.2 static void bt3c_forward (node_p, hook_p, void *, int) [static]

Definition at line 1015 of file ng_bt3c_pccard.c.

References `bt3c_softc::hook`, `bt3c_softc::inq`, `NG_BT3C_STAT_IERROR`, `NG_FREE_M`, `NG_HOOK_IS_INVALID`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, and `bt3c_softc::stat`.

Referenced by `bt3c_swi_intr()`.

7.23.2.3 `static void bt3c_intr (void *) [static]`

Definition at line 754 of file `ng_bt3c_pccard.c`.

References `bt3c_read`, `bt3c_read_control`, `bt3c_receive()`, `bt3c_write`, `bt3c_write_control`, `bt3c_softc::dev`, `bt3c_softc::ith`, `NG_BT3C_INFO`, `NG_BT3C_NODE_TYPE`, and `NG_BT3C_WARN`.

Referenced by `bt3c_pccard_attach()`.

Here is the call graph for this function:

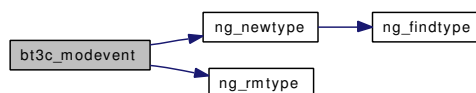


7.23.2.4 `static int bt3c_modevent (module_t mod, int event, void * data) [static]`

Definition at line 1201 of file `ng_bt3c_pccard.c`.

References `NG_BT3C_NODE_TYPE`, `ng_newtype()`, `ng_rmtype()`, and `typestruct`.

Here is the call graph for this function:

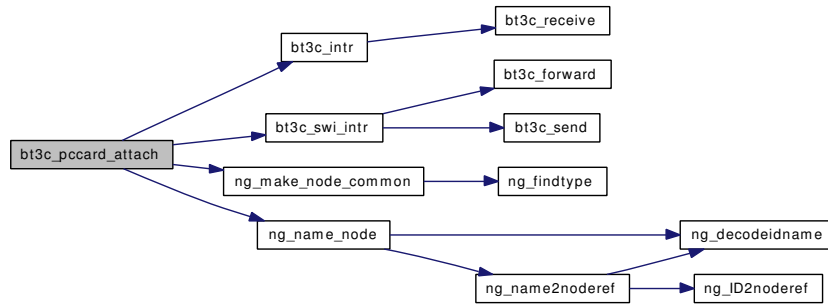


7.23.2.5 `static int bt3c_pccard_attach (device_t) [static]`

Definition at line 613 of file `ng_bt3c_pccard.c`.

References `BT3C_DEFAULTQLEN`, `bt3c_intr()`, `bt3c_swi_intr()`, `bt3c_softc::debug`, `bt3c_softc::dev`, `bt3c_softc::inq`, `bt3c_softc::iobase`, `bt3c_softc::iobase_rid`, `bt3c_softc::ioh`, `bt3c_softc::iot`, `bt3c_softc::irq`, `bt3c_softc::irq_cookie`, `bt3c_softc::irq_rid`, `bt3c_softc::ith`, `NG_BT3C_W4_PKT_IND`, `NG_BT3C_WARN_LEVEL`, `ng_make_node_common()`, `ng_name_node()`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `bt3c_softc::node`, `bt3c_softc::outq`, `bt3c_softc::state`, `typestruct`, and `bt3c_softc::want`.

Here is the call graph for this function:

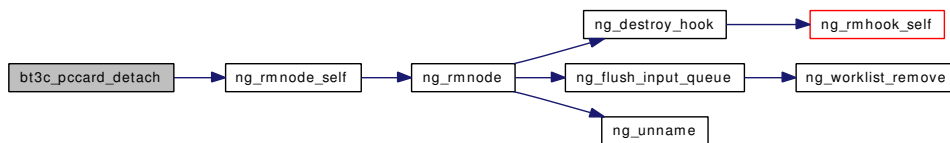


7.23.2.6 static int bt3c_pccard_detach (device_t) [static]

Definition at line 713 of file ng_bt3c_pccard.c.

References bt3c_softc::inq, bt3c_softc::iobase, bt3c_softc::iobase_rid, bt3c_softc::irq, bt3c_softc::irq_cookie, bt3c_softc::irq_rid, bt3c_softc::ith, bt3c_softc::m, NG_FREE_M, NG_NODE_SET_PRIVATE, ng_rmnode_self(), bt3c_softc::node, and bt3c_softc::outq.

Here is the call graph for this function:



7.23.2.7 static int bt3c_pccard_probe (device_t) [static]

Definition at line 589 of file ng_bt3c_pccard.c.

7.23.2.8 static void bt3c_receive (bt3c_softc_p) [static]

Definition at line 794 of file ng_bt3c_pccard.c.

References bt3c_read, bt3c_set_address, bt3c_softc::dev, bt3c_softc::m, NG_BT3C_ERR, NG_BT3C_INFO, NG_BT3C_STAT_IERROR, NG_BT3C_W4_PKT_IND, NG_FREE_M, bt3c_softc::stat, bt3c_softc::state, and bt3c_softc::want.

Referenced by bt3c_intr().

7.23.2.9 static void bt3c_send (node_p, hook_p, void *, int) [static]

Definition at line 1053 of file ng_bt3c_pccard.c.

References BT3C_FIFO_SIZE, bt3c_set_address, bt3c_write, bt3c_write_data, BT3C_XMIT, bt3c_softc::dev, bt3c_softc::flags, min, NG_BT3C_INFO, NG_BT3C_STAT_BYTES_SENT, NG_BT3C_STAT_PCKTS_SENT, NG_NODE_PRIVATE, bt3c_softc::outq, and bt3c_softc::stat.

Referenced by bt3c_swi_intr(), and ng_bt3c_rcvdata().

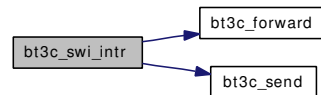
7.23.2.10 static void bt3c_swi_intr (void *) [static]

Definition at line 973 of file ng_bt3c_pccard.c.

References BT3C_ANTENNA_OUT, bt3c_forward(), bt3c_read, bt3c_send(), bt3c_softc::dev, bt3c_softc::flags, NG_BT3C_ALERT, NG_BT3C_INFO, ng_send_fn, bt3c_softc::node, and bt3c_softc::status.

Referenced by bt3c_pccard_attach().

Here is the call graph for this function:

**7.23.2.11 DRIVER_MODULE (bt3c, pccard, bt3c_pccard_driver, bt3c_devclass, bt3c_modevent, 0)****7.23.2.12 static MALLOC_DEFINE (M_BT3C, "bt3c", "bt3c data structures") [static]****7.23.2.13 MODULE_DEPEND (ng_bt3c, netgraph, NG_ABI_VERSION, NG_ABI_VERSION, NG_ABI_VERSION)****7.23.2.14 MODULE_VERSION (ng_bt3c, NG_BLUETOOTH_VERSION)****7.23.2.15 static int ng_bt3c_connect (hook_p hook) [static]**

Definition at line 315 of file ng_bt3c_pccard.c.

References bt3c_softc::hook, NG_HOOK_FORCE_QUEUE, NG_HOOK_NODE, NG_HOOK_PEER, and NG_NODE_PRIVATE.

7.23.2.16 static int ng_bt3c_constructor (node_p node) [static]

Definition at line 248 of file ng_bt3c_pccard.c.

7.23.2.17 static int ng_bt3c_disconnect (hook_p hook) [static]

Definition at line 335 of file ng_bt3c_pccard.c.

References bt3c_softc::hook, bt3c_softc::inq, NG_HOOK_NODE, NG_NODE_PRIVATE, and bt3c_softc::outq.

7.23.2.18 static int ng_bt3c_newhook (node_p node, hook_p hook, char const * name) [static]

Definition at line 295 of file ng_bt3c_pccard.c.

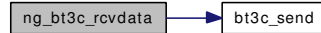
References bt3c_softc::hook, NG_BT3C_HOOK, and NG_NODE_PRIVATE.

7.23.2.19 static int ng_bt3c_rcvdata (**hook_p** hook, **item_p** item) [static]

Definition at line 540 of file ng_bt3c_pccard.c.

References bt3c_send(), bt3c_softc::dev, bt3c_softc::hook, NG_BT3C_ERR, NG_BT3C_STAT_OERROR, NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_send_fn, NGI_GET_M, bt3c_softc::node, bt3c_softc::outq, and bt3c_softc::stat.

Here is the call graph for this function:

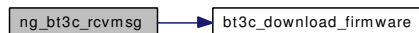


7.23.2.20 static int ng_bt3c_rcvmsg (**node_p** node, **item_p** item, **hook_p** lasthook) [static]

Definition at line 362 of file ng_bt3c_pccard.c.

References ng_mesg::ng_msghdr::arglen, bt3c_download_firmware(), ng_mesg::ng_msghdr::cmd, ng_mesg::data, bt3c_softc::debug, bt3c_softc::flags, ng_mesg::header, bt3c_softc::hook, bt3c_softc::inq, NG_BT3C_HOOK, NG_BT3C_STAT_RESET, NG_FREE_ITEM, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NG_TEXTRESPONSE, NGI_GET_MSG, NGM_BT3C_COOKIE, NGM_BT3C_NODE_DOWNLOAD_FIRMWARE, NGM_BT3C_NODE_GET_DEBUG, NGM_BT3C_NODE_GET_QLEN, NGM_BT3C_NODE_GET_STAT, NGM_BT3C_NODE_GET_STATE, NGM_BT3C_NODE_IN_QUEUE, NGM_BT3C_NODE_OUT_QUEUE, NGM_BT3C_NODE_RESET_STAT, NGM_BT3C_NODE_SET_DEBUG, NGM_BT3C_NODE_SET_QLEN, NGM_GENERIC_COOKIE, NGM_TEXT_STATUS, bt3c_softc::outq, bt3c_softc::stat, bt3c_softc::state, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:

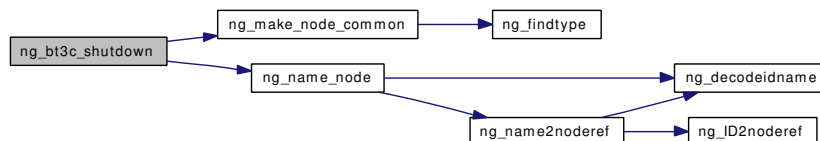


7.23.2.21 static int ng_bt3c_shutdown (**node_p** node) [static]

Definition at line 258 of file ng_bt3c_pccard.c.

References bt3c_softc::dev, ng_make_node_common(), ng_name_node(), NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, bt3c_softc::node, and typestruct.

Here is the call graph for this function:



7.23.3 Variable Documentation

7.23.3.1 `devclass_t bt3c_devclass` [static]

Definition at line 1193 of file `ng_bt3c_pccard.c`.

7.23.3.2 `driver_t bt3c_pccard_driver` [static]

Initial value:

```
{
    NG_BT3C_NODE_TYPE,
    bt3c_pccard_methods,
    sizeof(bt3c_softc_t)
}
```

Definition at line 1187 of file `ng_bt3c_pccard.c`.

7.23.3.3 `device_method_t bt3c_pccard_methods[]` [static]

Initial value:

```
{
    DEVMETHOD(device_probe,      bt3c_pccard_probe),
    DEVMETHOD(device_attach,    bt3c_pccard_attach),
    DEVMETHOD(device_detach,    bt3c_pccard_detach),
    { 0, 0 }
}
```

Definition at line 1178 of file `ng_bt3c_pccard.c`.

7.23.3.4 `struct ng_cmdlist ng_bt3c_cmdlist[]` [static]

Definition at line 177 of file `ng_bt3c_pccard.c`.

7.23.3.5 `ng_connect_t ng_bt3c_connect` [static]

Definition at line 78 of file `ng_bt3c_pccard.c`.

7.23.3.6 `ng_constructor_t ng_bt3c_constructor` [static]

Definition at line 75 of file `ng_bt3c_pccard.c`.

7.23.3.7 `ng_disconnect_t ng_bt3c_disconnect` [static]

Definition at line 79 of file `ng_bt3c_pccard.c`.

7.23.3.8 `ng_newhook_t ng_bt3c_newhook` [static]

Definition at line 77 of file `ng_bt3c_pccard.c`.

7.23.3.9 struct [ng_parse_type ng_bt3c_node_qlen_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bt3c_node_qlen_type_fields
}
```

Definition at line 156 of file `ng_bt3c_pccard.c`.

7.23.3.10 struct [ng_parse_struct_field ng_bt3c_node_qlen_type_fields\[\]](#) [static]**Initial value:**

```
{
    { "queue", &ng_parse_int32_type, },
    { "qlen", &ng_parse_int32_type, },
    { NULL, }
}
```

Definition at line 150 of file `ng_bt3c_pccard.c`.

7.23.3.11 struct [ng_parse_type ng_bt3c_node_stat_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bt3c_node_stat_type_fields
}
```

Definition at line 172 of file `ng_bt3c_pccard.c`.

7.23.3.12 struct [ng_parse_struct_field ng_bt3c_node_stat_type_fields\[\]](#) [static]**Initial value:**

```
{
    { "pkts_recv", &ng_parse_uint32_type, },
    { "bytes_recv", &ng_parse_uint32_type, },
    { "pkts_sent", &ng_parse_uint32_type, },
    { "bytes_sent", &ng_parse_uint32_type, },
    { "oerrors", &ng_parse_uint32_type, },
    { "ierrors", &ng_parse_uint32_type, },
    { NULL, }
}
```

Definition at line 162 of file `ng_bt3c_pccard.c`.

7.23.3.13 [ng_rcvdata_t ng_bt3c_rcvdata](#) [static]

Definition at line 81 of file `ng_bt3c_pccard.c`.

7.23.3.14 `ng_rcvmsg_t ng_bt3c_rcvmsg` [static]

Definition at line 80 of file `ng_bt3c_pccard.c`.

7.23.3.15 `ng_shutdown_t ng_bt3c_shutdown` [static]

Definition at line 76 of file `ng_bt3c_pccard.c`.

7.23.3.16 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_BT3C_NODE_TYPE,
    .constructor = ng_bt3c_constructor,
    .rcvmsg =      ng_bt3c_rcvmsg,
    .shutdown =    ng_bt3c_shutdown,
    .newhook =     ng_bt3c_newhook,
    .connect =     ng_bt3c_connect,
    .rcvdata =     ng_bt3c_rcvdata,
    .disconnect =  ng_bt3c_disconnect,
    .cmdlist =     ng_bt3c_cmdlist
}
```

Definition at line 230 of file `ng_bt3c_pccard.c`.

Referenced by `bt3c_modevent()`, `bt3c_pccard_attach()`, `ng_attach_cntl()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_ubt_shutdown()`, `ngt_open()`, and `ubt_modevent()`.

7.24 /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [bt3c_softc](#)

Defines

- #define [NG_BT3C_ALERT](#) if (sc → debug >= NG_BT3C_ALERT_LEVEL) device_printf
- #define [NG_BT3C_ERR](#) if (sc → debug >= NG_BT3C_ERR_LEVEL) device_printf
- #define [NG_BT3C_WARN](#) if (sc → debug >= NG_BT3C_WARN_LEVEL) device_printf
- #define [NG_BT3C_INFO](#) if (sc → debug >= NG_BT3C_INFO_LEVEL) device_printf
- #define [BT3C_DATA_L](#) 0x00
- #define [BT3C_DATA_H](#) 0x01
- #define [BT3C_ADDR_L](#) 0x02
- #define [BT3C_ADDR_H](#) 0x03
- #define [BT3C_CONTROL](#) 0x04
- #define [BT3C_FIFO_SIZE](#) 256
- #define [BT3C_ANTENNA_OUT](#) (1 << 0)
- #define [BT3C_XMIT](#) (1 << 1)
- #define [NG_BT3C_STAT_PCKTS_SENT](#)(s) (s).pckts_sent ++
- #define [NG_BT3C_STAT_BYTES_SENT](#)(s, n) (s).bytes_sent += (n)
- #define [NG_BT3C_STAT_PCKTS_RECV](#)(s) (s).pckts_recv ++
- #define [NG_BT3C_STAT_BYTES_RECV](#)(s, n) (s).bytes_recv += (n)
- #define [NG_BT3C_STAT_OERROR](#)(s) (s).oerrors ++
- #define [NG_BT3C_STAT_IERROR](#)(s) (s).ierrors ++
- #define [NG_BT3C_STAT_RESET](#)(s) bzero(&(s), sizeof((s)))
- #define [BT3C_DEFAULTQLEN](#) 12

Typedefs

- typedef [bt3c_softc](#) [bt3c_softc_t](#)
- typedef [bt3c_softc](#) * [bt3c_softc_p](#)

7.24.1 Define Documentation

7.24.1.1 #define BT3C_ADDR_H 0x03

Definition at line 55 of file [ng_bt3c_var.h](#).

7.24.1.2 #define BT3C_ADDR_L 0x02

Definition at line 54 of file ng_bt3c_var.h.

7.24.1.3 #define BT3C_ANTENNA_OUT (1 << 0)

Definition at line 78 of file ng_bt3c_var.h.

Referenced by bt3c_swi_intr().

7.24.1.4 #define BT3C_CONTROL 0x04

Definition at line 56 of file ng_bt3c_var.h.

7.24.1.5 #define BT3C_DATA_H 0x01

Definition at line 53 of file ng_bt3c_var.h.

7.24.1.6 #define BT3C_DATA_L 0x00

Definition at line 52 of file ng_bt3c_var.h.

7.24.1.7 #define BT3C_DEFAULTQLEN 12

Definition at line 100 of file ng_bt3c_var.h.

Referenced by bt3c_pccard_attach().

7.24.1.8 #define BT3C_FIFO_SIZE 256

Definition at line 58 of file ng_bt3c_var.h.

Referenced by bt3c_send().

7.24.1.9 #define BT3C_XMIT (1 << 1)

Definition at line 79 of file ng_bt3c_var.h.

Referenced by bt3c_send().

**7.24.1.10 #define NG_BT3C_ALERT if (sc → debug >= NG_BT3C_ALERT_LEVEL)
device_printf**

Definition at line 46 of file ng_bt3c_var.h.

Referenced by bt3c_swi_intr().

7.24.1.11 #define NG_BT3C_ERR if (sc → debug >= NG_BT3C_ERR_LEVEL) device_printf

Definition at line 47 of file ng_bt3c_var.h.

Referenced by bt3c_receive(), and ng_bt3c_rcvdata().

7.24.1.12 #define NG_BT3C_INFO if (sc → debug >= NG_BT3C_INFO_LEVEL) device_printf

Definition at line 49 of file ng_bt3c_var.h.

Referenced by bt3c_intr(), bt3c_receive(), bt3c_send(), and bt3c_swi_intr().

7.24.1.13 #define NG_BT3C_STAT_BYTES_RECV(s, n) (s).bytes_recv += (n)

Definition at line 87 of file ng_bt3c_var.h.

7.24.1.14 #define NG_BT3C_STAT_BYTES_SENT(s, n) (s).bytes_sent += (n)

Definition at line 85 of file ng_bt3c_var.h.

Referenced by bt3c_send().

7.24.1.15 #define NG_BT3C_STAT_IERROR(s) (s).ierrors ++

Definition at line 89 of file ng_bt3c_var.h.

Referenced by bt3c_forward(), and bt3c_receive().

7.24.1.16 #define NG_BT3C_STAT_OERROR(s) (s).oerrors ++

Definition at line 88 of file ng_bt3c_var.h.

Referenced by ng_bt3c_rcvdata().

7.24.1.17 #define NG_BT3C_STAT_PCKTS_RECV(s) (s).pckts_recv ++

Definition at line 86 of file ng_bt3c_var.h.

7.24.1.18 #define NG_BT3C_STAT_PCKTS_SENT(s) (s).pckts_sent ++

Definition at line 84 of file ng_bt3c_var.h.

Referenced by bt3c_send().

7.24.1.19 #define NG_BT3C_STAT_RESET(s) bzero(&(s), sizeof((s)))

Definition at line 90 of file ng_bt3c_var.h.

Referenced by ng_bt3c_rcvmsg().

7.24.1.20 #define NG_BT3C_WARN if (sc → debug >= NG_BT3C_WARN_LEVEL) device_printf

Definition at line 48 of file ng_bt3c_var.h.

Referenced by bt3c_intr().

7.24.2 Typedef Documentation**7.24.2.1 typedef struct bt3c_softc* bt3c_softc_p**

Definition at line 104 of file ng_bt3c_var.h.

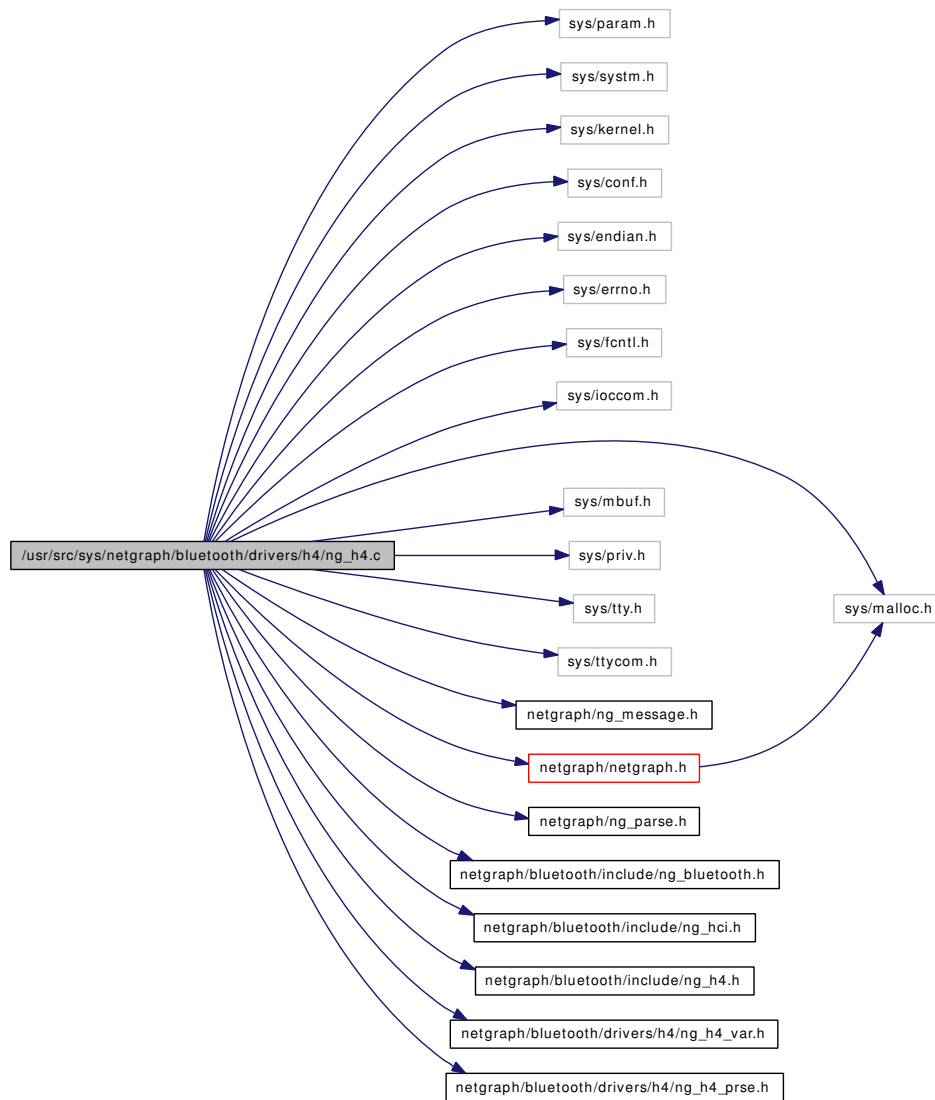
7.24.2.2 typedef struct bt3c_softc bt3c_softc_t

Definition at line 103 of file ng_bt3c_var.h.

7.25 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/conf.h>
#include <sys/endian.h>
#include <sys/errno.h>
#include <sys/fcntl.h>
#include <sys/ioccom.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/priv.h>
#include <sys/tty.h>
#include <sys/ttycom.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_h4.h>
#include <netgraph/bluetooth/drivers/h4/ng_h4_var.h>
#include <netgraph/bluetooth/drivers/h4/ng_h4_prse.h>
```

Include dependency graph for ng_h4.c:



Defines

- `#define NI(x) ((struct nodeinfo *) (x))`

Functions

- `NET_NEEDS_GIANT` ("ng_h4")
- `MALLOC_DEFINE` (M_NETGRAPH_H4, "netgraph_h4", "Netgraph Bluetooth H4 node")
- static int `ng_h4_open` (struct cdev *, struct tty *)
- static int `ng_h4_close` (struct tty *, int)
- static int `ng_h4_read` (struct tty *, struct uio *, int)
- static int `ng_h4_write` (struct tty *, struct uio *, int)
- static int `ng_h4_input` (int, struct tty *)
- static int `ng_h4_start` (struct tty *)
- static void `ng_h4_start2` (node_p, hook_p, void *, int)

- static int `ng_h4_ioctl` (struct tty *, u_long, caddr_t, int, struct thread *)
- static void `ng_h4_timeout` (node_p)
- static void `ng_h4_untimeout` (node_p)
- static void `ng_h4_process_timeout` (node_p, hook_p, void *, int)
- static int `ng_h4_mod_event` (module_t, int, void *)
- `NETGRAPH_INIT` (h4, &typestruct)
- `MODULE_VERSION` (ng_h4, NG_BLUETOOTH_VERSION)
- static int `ng_h4_constructor` (node_p node)
- static int `ng_h4_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_h4_connect` (hook_p hook)
- static int `ng_h4_disconnect` (hook_p hook)
- static int `ng_h4_shutdown` (node_p node)
- static int `ng_h4_rcvdata` (hook_p hook, item_p item)
- static int `ng_h4_rcvmsg` (node_p node, item_p item, hook_p lasthook)

Variables

- static struct linesw `ng_h4_disc`
- static `ng_constructor_t` `ng_h4_constructor`
- static `ng_rcvmsg_t` `ng_h4_rcvmsg`
- static `ng_shutdown_t` `ng_h4_shutdown`
- static `ng_newhook_t` `ng_h4_newhook`
- static `ng_connect_t` `ng_h4_connect`
- static `ng_rcvdata_t` `ng_h4_rcvdata`
- static `ng_disconnect_t` `ng_h4_disconnect`
- static struct `ng_type` `typestruct`
- static int `ng_h4_node` = 0

7.25.1 Define Documentation

7.25.1.1 #define NI(x) ((struct nodeinfo *)(x))

Referenced by `ng_h4_ioctl()`.

7.25.2 Function Documentation

7.25.2.1 MALLOC_DEFINE (M_NETGRAPH_H4, "netgraph_h4", "Netgraph Bluetooth H4 node")

7.25.2.2 MODULE_VERSION (ng_h4, NG_BLUETOOTH_VERSION)

7.25.2.3 NET_NEEDS_GIANT ("ng_h4")

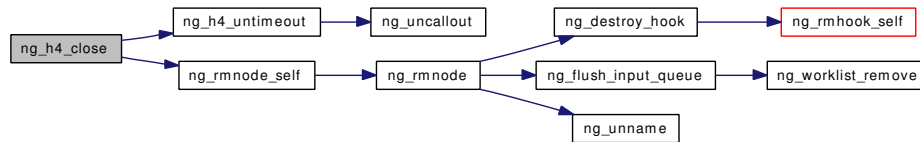
7.25.2.4 NETGRAPH_INIT (h4, & typestruct)

7.25.2.5 static int ng_h4_close (struct tty *, int) [static]

Definition at line 233 of file `ng_h4.c`.

References `ng_h4_info::flags`, `NG_BT_MBUFQ_DESTROY`, `NG_H4_TIMEOUT`, `ng_h4_untimeout()`, `NG_NODE_SET_PRIVATE`, `ng_rmnode_self()`, `ng_h4_info::node`, and `ng_h4_info::outq`.

Here is the call graph for this function:



7.25.2.6 `static int ng_h4_connect (hook_p hook)` [static]

Definition at line 673 of file `ng_h4.c`.

References `ng_h4_info::hook`, `NG_HOOK_FORCE_QUEUE`, `NG_HOOK_NODE`, `NG_HOOK_PEER`, and `NG_NODE_PRIVATE`.

7.25.2.7 `static int ng_h4_constructor (node_p node)` [static]

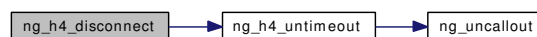
Definition at line 643 of file `ng_h4.c`.

7.25.2.8 `static int ng_h4_disconnect (hook_p hook)` [static]

Definition at line 692 of file `ng_h4.c`.

References `ng_h4_info::flags`, `ng_h4_info::got`, `ng_h4_info::hook`, `NG_BT_MBUFQ_DRAIN`, `NG_H4_TIMEOUT`, `ng_h4_untimeout()`, `NG_H4_W4_PKT_IND`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `ng_h4_info::outq`, `ng_h4_info::state`, and `ng_h4_info::want`.

Here is the call graph for this function:



7.25.2.9 `static int ng_h4_input (int, struct tty *)` [static]

Definition at line 331 of file `ng_h4.c`.

References `ng_h4_info::got`, `ng_h4_info::hook`, `ng_h4_info::ibuf`, `min`, `NG_H4_ALERT`, `NG_H4_ERR`, `NG_H4_INFO`, `NG_H4_STAT_BYTES_RECV`, `NG_H4_STAT_IERROR`, `NG_H4_STAT_PCKTS_RECV`, `NG_H4_W4_PKT_DATA`, `NG_H4_W4_PKT_HDR`, `NG_H4_W4_PKT_IND`, `NG_H4_WARN`, `NG_HCI_ACL_DATA_PKT`, `NG_HCI_EVENT_PKT`, `NG_HCI_SCO_DATA_PKT`, `NG_HOOK_IS_VALID`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_SEND_DATA_ONLY`, `ng_h4_info::node`, `ng_h4_info::stat`, `ng_h4_info::state`, `ng_h4_info::tp`, and `ng_h4_info::want`.

7.25.2.10 `static int ng_h4_ioctl (struct tty *, u_long, caddr_t, int, struct thread *)` [static]

Definition at line 289 of file `ng_h4.c`.

References `ng_type::name`, `name`, `ng_node::nd_type`, `ng_node2ID()`, `NG_NODE_HAS_NAME`, `NG_NODE_NAME`, `NG_NODE_NUMHOOKS`, `NGIOCGINFO`, `NI`, and `ng_h4_info::node`.

Here is the call graph for this function:



7.25.2.11 `static int ng_h4_mod_event (module_t, int, void *) [static]`

Definition at line 1003 of file `ng_h4.c`.

References `ng_h4_disc`.

7.25.2.12 `static int ng_h4_newhook (node_p node, hook_p hook, const char * name) [static]`

Definition at line 653 of file `ng_h4.c`.

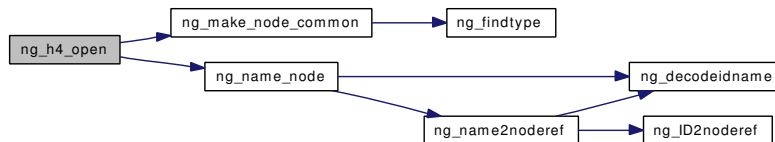
References `ng_h4_info::hook`, `NG_H4_HOOK`, and `NG_NODE_PRIVATE`.

7.25.2.13 `static int ng_h4_open (struct cdev *, struct tty *) [static]`

Definition at line 153 of file `ng_h4.c`.

References `ng_type::name`, `name`, `NG_BT_MBUFQ_INIT`, `ng_callout_init`, `NG_H4_ALERT`, `NG_H4_DEFAULTQLEN`, `NG_H4_HIWATER`, `ng_h4_node`, `NG_H4_W4_PKT_IND`, `NG_H4_WARN_LEVEL`, `ng_make_node_common()`, `ng_name_node()`, `NG_NODE_FORCE_WRITER`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `NG_NODESIZ`, and `typestruct`.

Here is the call graph for this function:



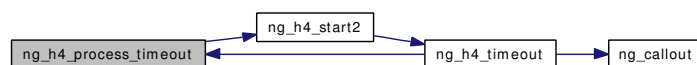
7.25.2.14 `static void ng_h4_process_timeout (node_p, hook_p, void *, int) [static]`

Definition at line 984 of file `ng_h4.c`.

References `ng_h4_info::flags`, `ng_h4_start2()`, `NG_H4_TIMEOUT`, and `NG_NODE_PRIVATE`.

Referenced by `ng_h4_timeout()`.

Here is the call graph for this function:

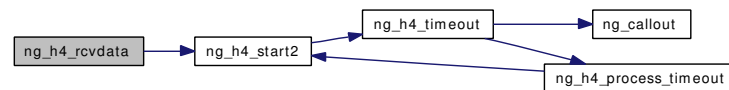


7.25.2.15 `static int ng_h4_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 769 of file ng_h4.c.

References `ng_h4_info::hook`, `NG_BT_MBUFQ_DROP`, `NG_BT_MBUFQ_ENQUEUE`, `NG_BT_MBUFQ_FULL`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_H4_ERR`, `NG_H4_INFO`, `ng_h4_start2()`, `NG_H4_STAT_OERROR`, `NG_HOOK_NODE`, `NG_NODE_NAME`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `ng_h4_info::node`, `ng_h4_info::outq`, and `ng_h4_info::stat`.

Here is the call graph for this function:

**7.25.2.16** `static int ng_h4_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 820 of file ng_h4.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_h4_info::debug`, `ng_h4_info::flags`, `ng_h4_info::got`, `ng_mesg::header`, `ng_h4_info::hook`, `ng_bt_mbufq::maxlen`, `NG_BT_MBUFQ_DRAIN`, `NG_BT_MBUFQ_LEN`, `NG_FREE_MSG`, `NG_H4_HOOK`, `NG_H4_STAT_RESET`, `NG_H4_W4_PKT_IND`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NG_TEXTRESPONSE`, `NGI_GET_MSG`, `NGM_GENERIC_COOKIE`, `NGM_H4_COOKIE`, `NGM_H4_NODE_GET_DEBUG`, `NGM_H4_NODE_GET_QLEN`, `NGM_H4_NODE_GET_STAT`, `NGM_H4_NODE_GET_STATE`, `NGM_H4_NODE_RESET`, `NGM_H4_NODE_RESET_STAT`, `NGM_H4_NODE_SET_DEBUG`, `NGM_H4_NODE_SET_QLEN`, `NGM_TEXT_STATUS`, `ng_h4_info::outq`, `ng_h4_info::stat`, `ng_h4_info::state`, `ng_mesg::ng_msghdr::typecookie`, and `ng_h4_info::want`.

7.25.2.17 `static int ng_h4_read (struct tty *, struct uio *, int)` [static]

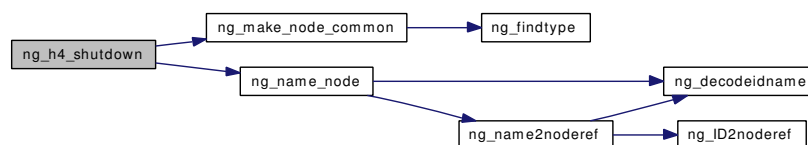
Definition at line 269 of file ng_h4.c.

7.25.2.18 `static int ng_h4_shutdown (node_p node)` [static]

Definition at line 726 of file ng_h4.c.

References `ng_type::name`, `name`, `ng_h4_node`, `ng_make_node_common()`, `ng_name_node()`, `NG_NODE_FORCE_WRITER`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `NG_NODESIZ`, `ng_h4_info::node`, and `typepstruct`.

Here is the call graph for this function:

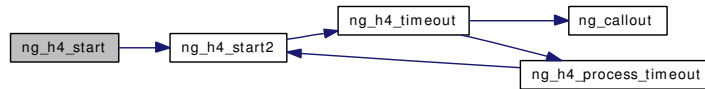


7.25.2.19 static int ng_h4_start (struct tty *) [static]

Definition at line 551 of file ng_h4.c.

References `ng_h4_start2()`, `NG_NODE_NOT_VALID`, `ng_send_fn`, `ng_h4_info::node`, and `ng_h4_info::tp`.

Here is the call graph for this function:



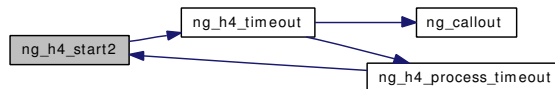
7.25.2.20 static void ng_h4_start2 (node_p, hook_p, void *, int) [static]

Definition at line 569 of file ng_h4.c.

References `ng_h4_info::flags`, `NG_BT_MBUFQ_DEQUEUE`, `NG_BT_MBUFQ_LEN`, `NG_BT_MBUFQ_PREPEND`, `NG_H4_HIWATER`, `NG_H4_STAT_BYTES_SENT`, `NG_H4_STAT_PCKTS_SENT`, `ng_h4_timeout()`, `NG_H4_TIMEOUT`, `NG_NODE_PRIVATE`, `ng_h4_info::outq`, `ng_h4_info::stat`, and `ng_h4_info::tp`.

Referenced by `ng_h4_process_timeout()`, `ng_h4_rcvdata()`, and `ng_h4_start()`.

Here is the call graph for this function:



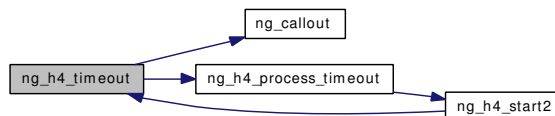
7.25.2.21 static void ng_h4_timeout (node_p) [static]

Definition at line 957 of file ng_h4.c.

References `ng_h4_info::flags`, `ng_callout()`, `ng_h4_process_timeout()`, `NG_H4_TIMEOUT`, `NG_NODE_PRIVATE`, and `ng_h4_info::timo`.

Referenced by `ng_h4_start2()`.

Here is the call graph for this function:



7.25.2.22 static void ng_h4_untimeout (node_p) [static]

Definition at line 970 of file ng_h4.c.

References `ng_h4_info::flags`, `NG_H4_TIMEOUT`, `NG_NODE_PRIVATE`, `ng_uncallout()`, and `ng_h4_info::timo`.

Referenced by `ng_h4_close()`, and `ng_h4_disconnect()`.

Here is the call graph for this function:



7.25.2.23 `static int ng_h4_write (struct tty *, struct uio *, int)` [static]

Definition at line 279 of file `ng_h4.c`.

7.25.3 Variable Documentation

7.25.3.1 `ng_connect_t ng_h4_connect` [static]

Definition at line 113 of file `ng_h4.c`.

7.25.3.2 `ng_constructor_t ng_h4_constructor` [static]

Definition at line 109 of file `ng_h4.c`.

7.25.3.3 `struct linesw ng_h4_disc` [static]

Initial value:

```

{
    ng_h4_open,
    ng_h4_close,
    ng_h4_read,
    ng_h4_write,
    ng_h4_ioctl,
    ng_h4_input,
    ng_h4_start,
    ttymodem
}
  
```

Definition at line 97 of file `ng_h4.c`.

Referenced by `ng_h4_mod_event()`.

7.25.3.4 `ng_disconnect_t ng_h4_disconnect` [static]

Definition at line 115 of file `ng_h4.c`.

7.25.3.5 `ng_newhook_t ng_h4_newhook` [static]

Definition at line 112 of file `ng_h4.c`.

7.25.3.6 int ng_h4_node = 0 [static]

Definition at line 140 of file ng_h4.c.

Referenced by ng_h4_open(), and ng_h4_shutdown().

7.25.3.7 ng_rcvdata_t ng_h4_rcvdata [static]

Definition at line 114 of file ng_h4.c.

7.25.3.8 ng_rcvmsg_t ng_h4_rcvmsg [static]

Definition at line 110 of file ng_h4.c.

7.25.3.9 ng_shutdown_t ng_h4_shutdown [static]

Definition at line 111 of file ng_h4.c.

7.25.3.10 struct ng_type typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_H4_NODE_TYPE,
    .mod_event =   ng_h4_mod_event,
    .constructor = ng_h4_constructor,
    .rcvmsg =      ng_h4_rcvmsg,
    .shutdown =    ng_h4_shutdown,
    .newhook =     ng_h4_newhook,
    .connect =     ng_h4_connect,
    .rcvdata =     ng_h4_rcvdata,
    .disconnect =  ng_h4_disconnect,
    .cmdlist =     ng_h4_cmdlist
}
```

Definition at line 124 of file ng_h4.c.

7.26 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_prse.h File Reference

This graph shows which files directly or indirectly include this file:



Variables

- static struct [ng_parse_struct_field ng_h4_stat_type_fields](#) []
- static struct [ng_parse_type ng_h4_stat_type](#)
- static struct [ng_cmdlist ng_h4_cmdlist](#) []

7.26.1 Variable Documentation

7.26.1.1 struct [ng_cmdlist ng_h4_cmdlist](#)[] [static]

Definition at line 63 of file [ng_h4_prse.h](#).

7.26.1.2 struct [ng_parse_type ng_h4_stat_type](#) [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_h4_stat_type_fields
}
  
```

Definition at line 58 of file [ng_h4_prse.h](#).

7.26.1.3 struct [ng_parse_struct_field ng_h4_stat_type_fields](#)[] [static]

Initial value:

```

{
    { "pkts_recv", &ng_parse_uint32_type, },
    { "bytes_recv", &ng_parse_uint32_type, },
    { "pkts_sent", &ng_parse_uint32_type, },
    { "bytes_sent", &ng_parse_uint32_type, },
    { "oerrors", &ng_parse_uint32_type, },
    { "ierrors", &ng_parse_uint32_type, },
    { NULL, }
}
  
```

Definition at line 48 of file [ng_h4_prse.h](#).

7.27 /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_h4_info](#)

Defines

- #define [NG_H4_ALERT](#) if (sc → debug >= NG_H4_ALERT_LEVEL) printf
- #define [NG_H4_ERR](#) if (sc → debug >= NG_H4_ERR_LEVEL) printf
- #define [NG_H4_WARN](#) if (sc → debug >= NG_H4_WARN_LEVEL) printf
- #define [NG_H4_INFO](#) if (sc → debug >= NG_H4_INFO_LEVEL) printf
- #define [NG_H4_HIWATER](#) 256
- #define [NG_H4_TIMEOUT](#) (1 << 0)
- #define [NG_H4_STAT_PCKTS_SENT](#)(s) (s).pckts_sent ++
- #define [NG_H4_STAT_BYTES_SENT](#)(s, n) (s).bytes_sent += (n)
- #define [NG_H4_STAT_PCKTS_RECV](#)(s) (s).pckts_recv ++
- #define [NG_H4_STAT_BYTES_RECV](#)(s, n) (s).bytes_recv += (n)
- #define [NG_H4_STAT_OERROR](#)(s) (s).oerrors ++
- #define [NG_H4_STAT_IERROR](#)(s) (s).ierrors ++
- #define [NG_H4_STAT_RESET](#)(s) bzero(&(s), sizeof((s)))
- #define [NG_H4_DEFAULTQLEN](#) 12
- #define [NG_H4_IBUF_SIZE](#) 1024

Typedefs

- typedef [ng_h4_info](#) [ng_h4_info_t](#)
- typedef [ng_h4_info_t](#) * [ng_h4_info_p](#)

Functions

- [MALLOC_DECLARE](#) (M_NETGRAPH_H4)

7.27.1 Define Documentation

7.27.1.1 #define NG_H4_ALERT if (sc → debug >= NG_H4_ALERT_LEVEL) printf

Definition at line 57 of file [ng_h4_var.h](#).

Referenced by [ng_h4_input\(\)](#), and [ng_h4_open\(\)](#).

7.27.1.2 #define NG_H4_DEFAULTQLEN 12

Definition at line 88 of file ng_h4_var.h.

Referenced by ng_h4_open().

7.27.1.3 #define NG_H4_ERR if (sc → debug >= NG_H4_ERR_LEVEL) printf

Definition at line 58 of file ng_h4_var.h.

Referenced by ng_h4_input(), and ng_h4_rcvdata().

7.27.1.4 #define NG_H4_HIWATER 256

Definition at line 62 of file ng_h4_var.h.

Referenced by ng_h4_open(), and ng_h4_start2().

7.27.1.5 #define NG_H4_IBUF_SIZE 1024

Definition at line 90 of file ng_h4_var.h.

7.27.1.6 #define NG_H4_INFO if (sc → debug >= NG_H4_INFO_LEVEL) printf

Definition at line 60 of file ng_h4_var.h.

Referenced by ng_h4_input(), and ng_h4_rcvdata().

7.27.1.7 #define NG_H4_STAT_BYTES_RECV(s, n) (s).bytes_recv += (n)

Definition at line 82 of file ng_h4_var.h.

Referenced by ng_h4_input().

7.27.1.8 #define NG_H4_STAT_BYTES_SENT(s, n) (s).bytes_sent += (n)

Definition at line 80 of file ng_h4_var.h.

Referenced by ng_h4_start2().

7.27.1.9 #define NG_H4_STAT_IERROR(s) (s).ierrors ++

Definition at line 84 of file ng_h4_var.h.

Referenced by ng_h4_input().

7.27.1.10 #define NG_H4_STAT_OERROR(s) (s).oerrors ++

Definition at line 83 of file ng_h4_var.h.

Referenced by ng_h4_rcvdata().

7.27.1.11 #define NG_H4_STAT_PCKTS_RECV(s) (s).pckts_recv ++

Definition at line 81 of file `ng_h4_var.h`.

Referenced by `ng_h4_input()`.

7.27.1.12 #define NG_H4_STAT_PCKTS_SENT(s) (s).pckts_sent ++

Definition at line 79 of file `ng_h4_var.h`.

Referenced by `ng_h4_start2()`.

7.27.1.13 #define NG_H4_STAT_RESET(s) bzero(&(s), sizeof((s)))

Definition at line 85 of file `ng_h4_var.h`.

Referenced by `ng_h4_rcvmsg()`.

7.27.1.14 #define NG_H4_TIMEOUT (1 << 0)

Definition at line 73 of file `ng_h4_var.h`.

Referenced by `ng_h4_close()`, `ng_h4_disconnect()`, `ng_h4_process_timeout()`, `ng_h4_start2()`, `ng_h4_timeout()`, and `ng_h4_untimeout()`.

7.27.1.15 #define NG_H4_WARN if (sc → debug >= NG_H4_WARN_LEVEL) printf

Definition at line 59 of file `ng_h4_var.h`.

Referenced by `ng_h4_input()`.

7.27.2 Typedef Documentation**7.27.2.1 typedef [ng_h4_info_t](#)* [ng_h4_info_p](#)**

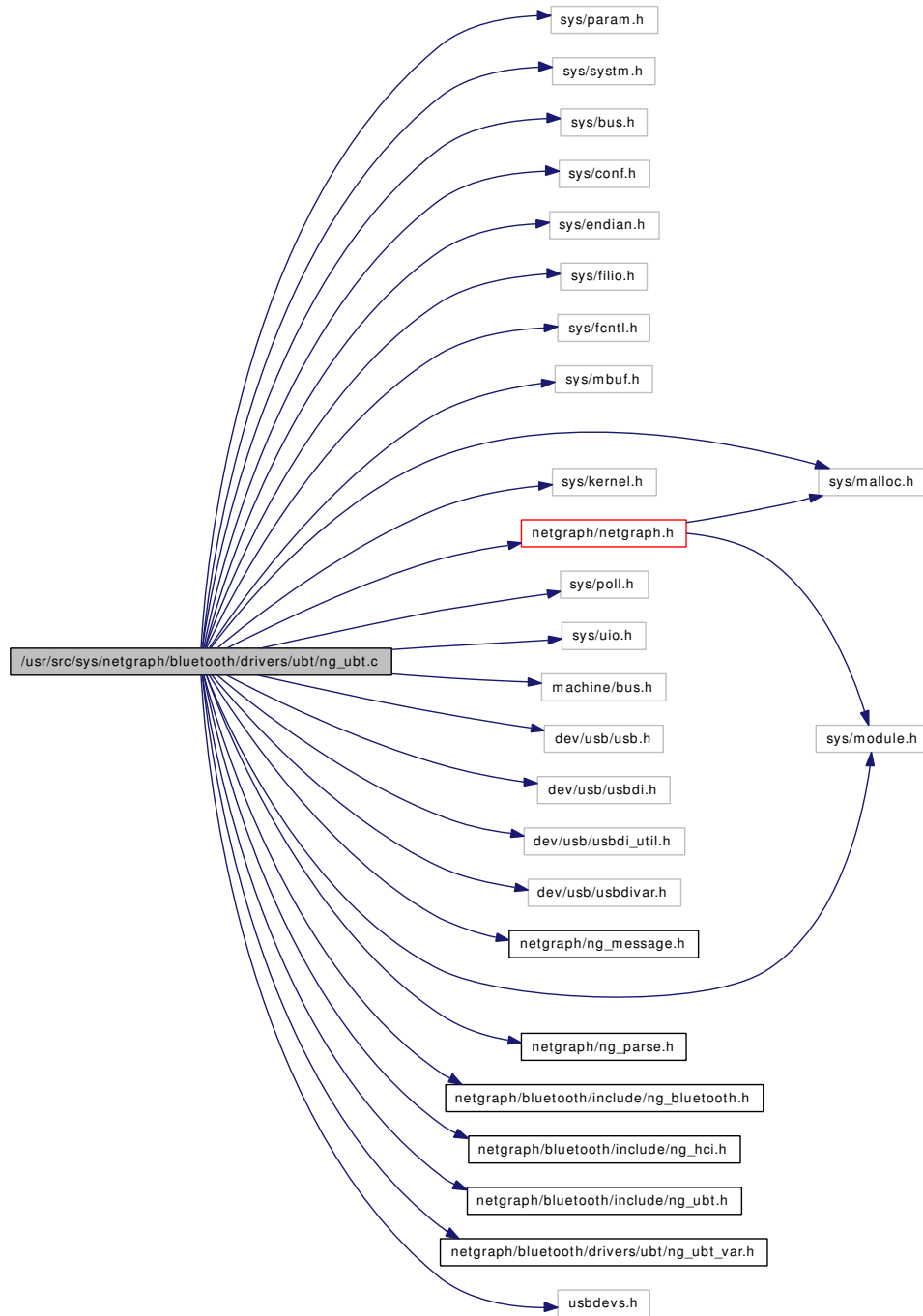
Definition at line 99 of file `ng_h4_var.h`.

7.27.2.2 typedef struct [ng_h4_info](#) [ng_h4_info_t](#)**7.27.3 Function Documentation****7.27.3.1 MALLOC_DECLARE (M_NETGRAPH_H4)**

7.28 /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bus.h>
#include <sys/conf.h>
#include <sys/endian.h>
#include <sys/filio.h>
#include <sys/fcntl.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/poll.h>
#include <sys/uio.h>
#include <machine/bus.h>
#include <dev/usb/usb.h>
#include <dev/usb/usbdi.h>
#include <dev/usb/usbdi_util.h>
#include <dev/usb/usbdivar.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_ubt.h>
#include <netgraph/bluetooth/drivers/ubt/ng_ubt_var.h>
#include "usbdevs.h"
```

Include dependency graph for ng_ubt.c:



Functions

- `USB_DECLARE_DRIVER` (ubt)
- `static int ubt_modevent` (module_t, int, void *)
- `static usbd_status ubt_request_start` (ubt_softc_p)
- `static void ubt_request_complete` (usbd_xfer_handle, usbd_private_handle, usbd_status)
- `static void ubt_request_complete2` (node_p, hook_p, void *, int)
- `static usbd_status ubt_intr_start` (ubt_softc_p)

- static void `ubt_intr_complete` (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)
- static void `ubt_intr_complete2` (node_p, hook_p, void *, int)
- static usb_d_status `ubt_bulk_in_start` (ubt_softc_p)
- static void `ubt_bulk_in_complete` (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)
- static void `ubt_bulk_in_complete2` (node_p, hook_p, void *, int)
- static usb_d_status `ubt_bulk_out_start` (ubt_softc_p)
- static void `ubt_bulk_out_complete` (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)
- static void `ubt_bulk_out_complete2` (node_p, hook_p, void *, int)
- static usb_d_status `ubt_isoc_in_start` (ubt_softc_p)
- static void `ubt_isoc_in_complete` (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)
- static void `ubt_isoc_in_complete2` (node_p, hook_p, void *, int)
- static usb_d_status `ubt_isoc_out_start` (ubt_softc_p)
- static void `ubt_isoc_out_complete` (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)
- static void `ubt_isoc_out_complete2` (node_p, hook_p, void *, int)
- static void `ubt_reset` (ubt_softc_p)
- `DRIVER_MODULE` (ubt, uhub, ubt_driver, ubt_devclass, ubt_modevent, 0)
- `MODULE_VERSION` (ng_ubt, NG_BLUETOOTH_VERSION)
- `MODULE_DEPEND` (ng_ubt, netgraph, NG_ABI_VERSION, NG_ABI_VERSION, NG_ABI_VERSION)
- `USB_MATCH` (ubt)
- `USB_ATTACH` (ubt)
- `USB_DETACH` (ubt)
- static int `ng_ubt_constructor` (node_p node)
- static int `ng_ubt_shutdown` (node_p node)
- static int `ng_ubt_newhook` (node_p node, hook_p hook, char const *name)
- static int `ng_ubt_connect` (hook_p hook)
- static int `ng_ubt_disconnect` (hook_p hook)
- static int `ng_ubt_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_ubt_rcvdata` (hook_p hook, item_p item)

Variables

- static `ng_constructor_t` `ng_ubt_constructor`
- static `ng_shutdown_t` `ng_ubt_shutdown`
- static `ng_newhook_t` `ng_ubt_newhook`
- static `ng_connect_t` `ng_ubt_connect`
- static `ng_disconnect_t` `ng_ubt_disconnect`
- static `ng_rcvmsg_t` `ng_ubt_rcvmsg`
- static `ng_rcvdata_t` `ng_ubt_rcvdata`
- static struct `ng_parse_struct_field` `ng_ubt_node_qlen_type_fields` []
- static struct `ng_parse_type` `ng_ubt_node_qlen_type`
- static struct `ng_parse_struct_field` `ng_ubt_node_stat_type_fields` []
- static struct `ng_parse_type` `ng_ubt_node_stat_type`
- static struct `ng_cmdlist` `ng_ubt_cmdlist` []
- static struct `ng_type` `typestruct`

7.28.1 Function Documentation

7.28.1.1 DRIVER_MODULE (*ubt*, *uhub*, *ubt_driver*, *ubt_devclass*, *ubt_modevent*, 0)

7.28.1.2 MODULE_DEPEND (*ng_ubt*, *netgraph*, *NG_ABI_VERSION*, *NG_ABI_VERSION*, *NG_ABI_VERSION*)

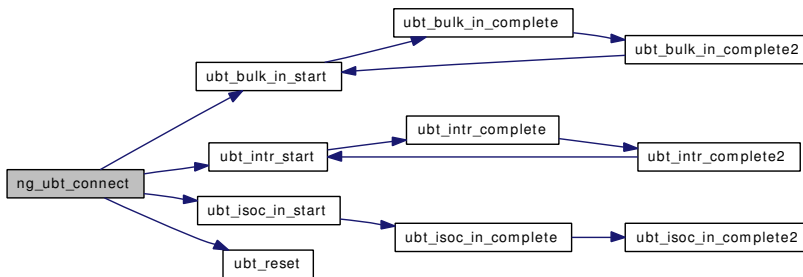
7.28.1.3 MODULE_VERSION (*ng_ubt*, *NG_BLUETOOTH_VERSION*)

7.28.1.4 static int ng_ubt_connect (*hook_p hook*) [static]

Definition at line 1882 of file *ng_ubt.c*.

References *NG_HOOK_FORCE_QUEUE*, *NG_HOOK_NODE*, *NG_HOOK_PEER*, *NG_NODE_PRIVATE*, *NG_UBT_ALERT*, *ubt_softc::sc_dev*, *ubt_softc::sc_hook*, *ubt_bulk_in_start()*, *ubt_intr_start()*, *ubt_isoc_in_start()*, and *ubt_reset()*.

Here is the call graph for this function:



7.28.1.5 static int ng_ubt_constructor (*node_p node*) [static]

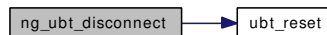
Definition at line 1813 of file *ng_ubt.c*.

7.28.1.6 static int ng_ubt_disconnect (*hook_p hook*) [static]

Definition at line 1934 of file *ng_ubt.c*.

References *NG_HOOK_NODE*, *NG_NODE_PRIVATE*, *ubt_softc::sc_hook*, and *ubt_reset()*.

Here is the call graph for this function:



7.28.1.7 static int ng_ubt_newhook (*node_p node*, *hook_p hook*, *char const * name*) [static]

Definition at line 1862 of file *ng_ubt.c*.

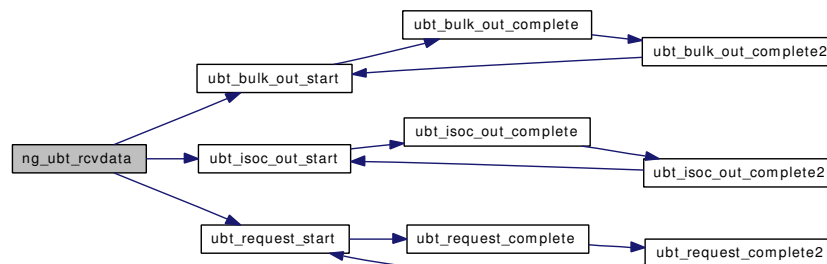
References *NG_NODE_PRIVATE*, *NG_UBT_HOOK*, and *ubt_softc::sc_hook*.

7.28.1.8 static int ng_ubt_rcvdata (hook_p hook, item_p item) [static]

Definition at line 2135 of file ng_ubt.c.

References NG_BT_MBUFQ_ENQUEUE, NG_BT_MBUFQ_FULL, NG_FREE_ITEM, NG_FREE_M, NG_HCI_ACL_DATA_PKT, NG_HCI_CMD_PKT, NG_HCI_SCO_DATA_PKT, NG_HOOK_NODE, NG_NODE_PRIVATE, NG_UBT_ERR, NGL_GET_M, ubt_softc::sc_aclq, ubt_softc::sc_cmdq, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_hook, ubt_softc::sc_scoq, UBT_ACL_XMIT, ubt_bulk_out_start(), UBT_CMD_XMIT, ubt_isoc_out_start(), UBT_NEED_FRAME_TYPE, ubt_request_start(), and UBT_SCO_XMIT.

Here is the call graph for this function:

**7.28.1.9 static int ng_ubt_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]**

Definition at line 1954 of file ng_ubt.c.

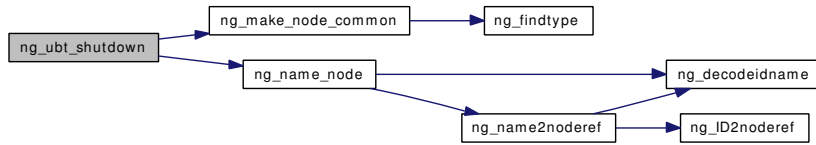
References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ng_mesg::header, ng_bt_mbufq::maxlen, NG_BT_MBUFQ_LEN, NG_FREE_ITEM, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NG_TEXTRESPONSE, NG_UBT_HOOK, NG_UBT_STAT_RESET, NGL_GET_MSG, NGM_GENERIC_COOKIE, NGM_TEXT_STATUS, NGM_UBT_COOKIE, NGM_UBT_NODE_GET_DEBUG, NGM_UBT_NODE_GET_QLEN, NGM_UBT_NODE_GET_STAT, NGM_UBT_NODE_QUEUE_ACL, NGM_UBT_NODE_QUEUE_CMD, NGM_UBT_NODE_QUEUE_SCO, NGM_UBT_NODE_RESET_STAT, NGM_UBT_NODE_SET_DEBUG, NGM_UBT_NODE_SET_QLEN, ubt_softc::sc_aclq, ubt_softc::sc_cmdq, ubt_softc::sc_debug, ubt_softc::sc_flags, ubt_softc::sc_hook, ubt_softc::sc_scoq, ubt_softc::sc_stat, and ng_mesg::ng_msghdr::typecookie.

7.28.1.10 static int ng_ubt_shutdown (node_p node) [static]

Definition at line 1823 of file ng_ubt.c.

References ng_make_node_common(), ng_name_node(), NG_NODE_FORCE_WRITER, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, ubt_softc::sc_dev, ubt_softc::sc_node, and typestruct.

Here is the call graph for this function:



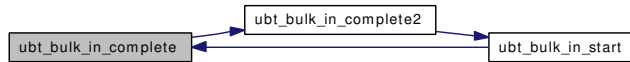
7.28.1.11 static void ubt_bulk_in_complete (usbd_xfer_handle, usbd_private_handle, usbd_status) [static]

Definition at line 1228 of file ng_ubt.c.

References NG_NODE_UNREF, ng_send_fn, and ubt_bulk_in_complete2().

Referenced by ubt_bulk_in_start().

Here is the call graph for this function:



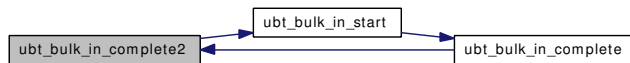
7.28.1.12 static void ubt_bulk_in_complete2 (node_p, hook_p, void *, int) [static]

Definition at line 1235 of file ng_ubt.c.

References NG_FREE_M, NG_HOOK_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, NG_UBT_ERR, NG_UBT_INFO, NG_UBT_STAT_BYTES_RECV, NG_UBT_STAT_IERROR, NG_UBT_STAT_PCKTS_RECV, NG_UBT_WARN, ubt_softc::sc_bulk_in_buffer, ubt_softc::sc_bulk_in_pipe, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_hook, ubt_softc::sc_stat, UBT_ACL_RECV, and ubt_bulk_in_start().

Referenced by ubt_bulk_in_complete().

Here is the call graph for this function:



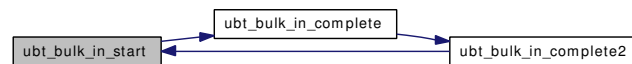
7.28.1.13 static usbd_status ubt_bulk_in_start (ubt_softc_p) [static]

Definition at line 1164 of file ng_ubt.c.

References NG_FREE_M, NG_HCI_ACL_DATA_PKT, NG_NODE_REF, NG_NODE_UNREF, NG_UBT_ERR, ubt_softc::sc_bulk_in_buffer, ubt_softc::sc_bulk_in_pipe, ubt_softc::sc_bulk_in_xfer, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_node, UBT_ACL_RECV, ubt_bulk_in_complete(), and UBT_HAVE_FRAME_TYPE.

Referenced by ng_ubt_connect(), and ubt_bulk_in_complete2().

Here is the call graph for this function:



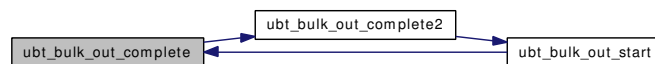
7.28.1.14 static void ubt_bulk_out_complete (usb_xfer_handle, usb_private_handle, usb_status) [static]

Definition at line 1405 of file ng_ubt.c.

References NG_NODE_UNREF, ng_send_fn, and ubt_bulk_out_complete2().

Referenced by ubt_bulk_out_start().

Here is the call graph for this function:



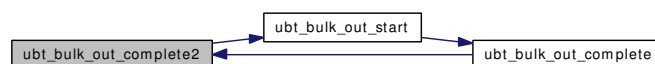
7.28.1.15 static void ubt_bulk_out_complete2 (node_p, hook_p, void *, int) [static]

Definition at line 1412 of file ng_ubt.c.

References NG_BT_MBUFQ_LEN, NG_NODE_PRIVATE, NG_UBT_INFO, NG_UBT_STAT_BYTES_SENT, NG_UBT_STAT_OERROR, NG_UBT_STAT_PCKTS_SENT, NG_UBT_WARN, ubt_softc::sc_aclq, ubt_softc::sc_bulk_out_pipe, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_stat, UBT_ACL_XMIT, and ubt_bulk_out_start().

Referenced by ubt_bulk_out_complete().

Here is the call graph for this function:



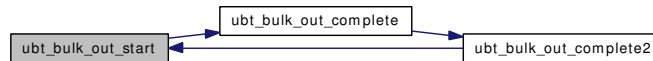
7.28.1.16 static usb_status ubt_bulk_out_start (ubt_softc_p) [static]

Definition at line 1330 of file ng_ubt.c.

References NG_BT_MBUFQ_DEQUEUE, NG_BT_MBUFQ_DROP, NG_FREE_M, NG_NODE_REF, NG_NODE_UNREF, NG_UBT_ERR, NG_UBT_INFO, NG_UBT_STAT_OERROR, ubt_softc::sc_aclq, ubt_softc::sc_bulk_out_buffer, ubt_softc::sc_bulk_out_pipe, ubt_softc::sc_bulk_out_xfer, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_node, ubt_softc::sc_stat, UBT_ACL_XMIT, UBT_BULK_BUFFER_SIZE, and ubt_bulk_out_complete().

Referenced by ng_ubt_rcvdata(), and ubt_bulk_out_complete2().

Here is the call graph for this function:



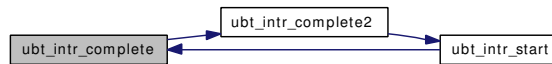
7.28.1.17 `static void ubt_intr_complete (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)`
`[static]`

Definition at line 1062 of file `ng_ubt.c`.

References `NG_NODE_UNREF`, `ng_send_fn`, and `ubt_intr_complete2()`.

Referenced by `ubt_intr_start()`.

Here is the call graph for this function:



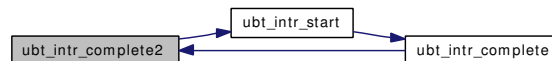
7.28.1.18 `static void ubt_intr_complete2 (node_p, hook_p, void *, int)` `[static]`

Definition at line 1069 of file `ng_ubt.c`.

References `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `NG_UBT_ERR`, `NG_UBT_INFO`, `NG_UBT_STAT_BYTES_RECV`, `NG_UBT_STAT_IERROR`, `NG_UBT_STAT_PCKTS_RECV`, `NG_UBT_WARN`, `ubt_softc::sc_dev`, `ubt_softc::sc_flags`, `ubt_softc::sc_hook`, `ubt_softc::sc_intr_buffer`, `ubt_softc::sc_intr_pipe`, `ubt_softc::sc_stat`, `UBT_EVT_RECV`, and `ubt_intr_start()`.

Referenced by `ubt_intr_complete()`.

Here is the call graph for this function:



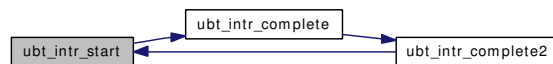
7.28.1.19 `static usb_d_status ubt_intr_start (ubt_softc_p)` `[static]`

Definition at line 998 of file `ng_ubt.c`.

References `NG_FREE_M`, `NG_HCI_EVENT_PKT`, `NG_NODE_REF`, `NG_NODE_UNREF`, `NG_UBT_ERR`, `ubt_softc::sc_dev`, `ubt_softc::sc_flags`, `ubt_softc::sc_intr_buffer`, `ubt_softc::sc_intr_pipe`, `ubt_softc::sc_intr_xfer`, `ubt_softc::sc_node`, `UBT_EVT_RECV`, `UBT_HAVE_FRAME_TYPE`, and `ubt_intr_complete()`.

Referenced by `ng_ubt_connect()`, and `ubt_intr_complete2()`.

Here is the call graph for this function:



7.28.1.20 static void ubt_isoc_in_complete (usb_d_xfer_handle, usb_d_private_handle, usb_d_status) [static]

Definition at line 1507 of file ng_ubt.c.

References NG_NODE_UNREF, ng_send_fn, and ubt_isoc_in_complete2().

Referenced by ubt_isoc_in_start().

Here is the call graph for this function:



7.28.1.21 static void ubt_isoc_in_complete2 (node_p, hook_p, void *, int) [static]

Definition at line 1514 of file ng_ubt.c.

References min, NG_HCI_SCO_DATA_PKT, NG_HOOK_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, NG_UBT_ALERT, NG_UBT_INFO, NG_UBT_STAT_BYTES_RECV, NG_UBT_STAT_IERROR, NG_UBT_STAT_PCKTS_RECV, NG_UBT_WARN, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_hook, ubt_softc::sc_isoc_in_buffer, ubt_softc::sc_isoc_in_frlen, ubt_softc::sc_isoc_in_pipe, ubt_softc::sc_isoc_nframes, ubt_softc::sc_isoc_size, ubt_softc::sc_stat, UBT_HAVE_FRAME_TYPE, and UBT_SCO_RECV.

Referenced by ubt_isoc_in_complete().

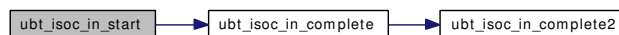
7.28.1.22 static usb_d_status ubt_isoc_in_start (ubt_softc_p) [static]

Definition at line 1461 of file ng_ubt.c.

References NG_NODE_REF, NG_NODE_UNREF, NG_UBT_ERR, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_isoc_in_frlen, ubt_softc::sc_isoc_in_pipe, ubt_softc::sc_isoc_in_xfer, ubt_softc::sc_isoc_nframes, ubt_softc::sc_isoc_size, ubt_softc::sc_node, ubt_isoc_in_complete(), and UBT_SCO_RECV.

Referenced by ng_ubt_connect().

Here is the call graph for this function:



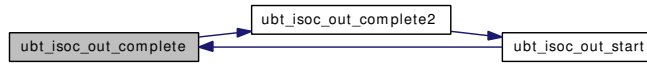
7.28.1.23 static void ubt_isoc_out_complete (usb_d_xfer_handle, usb_d_private_handle, usb_d_status) [static]

Definition at line 1721 of file ng_ubt.c.

References NG_NODE_UNREF, ng_send_fn, and ubt_isoc_out_complete2().

Referenced by ubt_isoc_out_start().

Here is the call graph for this function:



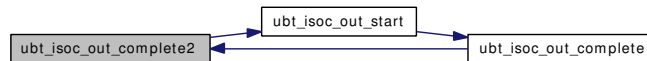
7.28.1.24 static void ubt_isoc_out_complete2 (node_p, hook_p, void *, int) [static]

Definition at line 1728 of file ng_ubt.c.

References NG_BT_MBUFQ_LEN, NG_NODE_PRIVATE, NG_UBT_INFO, NG_UBT_STAT_BYTES_SENT, NG_UBT_STAT_OERROR, NG_UBT_STAT_PCKTS_SENT, NG_UBT_WARN, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_isoc_out_pipe, ubt_softc::sc_scoq, ubt_softc::sc_stat, ubt_isoc_out_start(), and UBT_SCO_XMIT.

Referenced by ubt_isoc_out_complete().

Here is the call graph for this function:



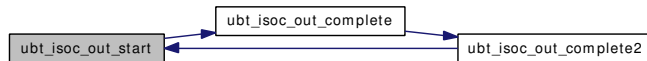
7.28.1.25 static usbd_status ubt_isoc_out_start (ubt_softc_p) [static]

Definition at line 1636 of file ng_ubt.c.

References min, NG_BT_MBUFQ_DEQUEUE, NG_BT_MBUFQ_DROP, NG_FREE_M, NG_NODE_REF, NG_NODE_UNREF, NG_UBT_ERR, NG_UBT_INFO, NG_UBT_STAT_OERROR, ubt_softc::sc_dev, ubt_softc::sc_flags, ubt_softc::sc_isoc_nframes, ubt_softc::sc_isoc_out_buffer, ubt_softc::sc_isoc_out_frlen, ubt_softc::sc_isoc_out_pipe, ubt_softc::sc_isoc_out_xfer, ubt_softc::sc_isoc_size, ubt_softc::sc_node, ubt_softc::sc_scoq, ubt_softc::sc_stat, ubt_isoc_out_complete(), and UBT_SCO_XMIT.

Referenced by ng_ubt_rcvdata(), and ubt_isoc_out_complete2().

Here is the call graph for this function:

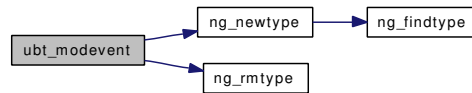


7.28.1.26 static int ubt_modevent (module_t, int, void *) [static]

Definition at line 224 of file ng_ubt.c.

References `ng_newtype()`, `ng_rmtime()`, `NG_UBT_NODE_TYPE`, and `typestruct`.

Here is the call graph for this function:



7.28.1.27 `static void ubt_request_complete (usb_d_xfer_handle, usb_d_private_handle, usb_d_status)` [static]

Definition at line 943 of file `ng_ubt.c`.

References `NG_NODE_UNREF`, `ng_send_fn`, and `ubt_request_complete2()`.

Referenced by `ubt_request_start()`.

Here is the call graph for this function:



7.28.1.28 `static void ubt_request_complete2 (node_p, hook_p, void *, int)` [static]

Definition at line 950 of file `ng_ubt.c`.

References `NG_BT_MBUFQ_LEN`, `NG_NODE_PRIVATE`, `NG_UBT_ERR`, `NG_UBT_INFO`, `NG_UBT_STAT_BYTES_SENT`, `NG_UBT_STAT_OERROR`, `NG_UBT_STAT_PCKTS_SENT`, `ubt_softc::sc_cmdq`, `ubt_softc::sc_dev`, `ubt_softc::sc_flags`, `ubt_softc::sc_stat`, `UBT_CMD_XMIT`, and `ubt_request_start()`.

Referenced by `ubt_request_complete()`.

Here is the call graph for this function:



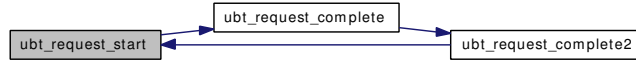
7.28.1.29 `static usb_d_status ubt_request_start (ubt_softc_p)` [static]

Definition at line 857 of file `ng_ubt.c`.

References `NG_BT_MBUFQ_DEQUEUE`, `NG_BT_MBUFQ_DROP`, `NG_FREE_M`, `NG_NODE_REF`, `NG_NODE_UNREF`, `NG_UBT_ERR`, `NG_UBT_INFO`, `NG_UBT_STAT_OERROR`, `ubt_softc::sc_cmdq`, `ubt_softc::sc_ctrl_buffer`, `ubt_softc::sc_ctrl_xfer`, `ubt_softc::sc_dev`, `ubt_softc::sc_flags`, `ubt_softc::sc_node`, `ubt_softc::sc_stat`, `ubt_softc::sc_udev`, `UBT_CMD_XMIT`, `UBT_CTRL_BUFFER_SIZE`, `UBT_HCI_REQUEST`, and `ubt_request_complete()`.

Referenced by `ng_ubt_rcvdata()`, and `ubt_request_complete2()`.

Here is the call graph for this function:



7.28.1.30 static void `ubt_reset(ubt_softc_p)` [static]

Definition at line 1778 of file `ng_ubt.c`.

References `NG_BT_MBUFQ_DRAIN`, `ubt_softc::sc_aclq`, `ubt_softc::sc_bulk_in_pipe`, `ubt_softc::sc_bulk_out_pipe`, `ubt_softc::sc_cmdq`, `ubt_softc::sc_intr_pipe`, `ubt_softc::sc_isoc_in_pipe`, `ubt_softc::sc_isoc_out_pipe`, and `ubt_softc::sc_scoq`.

Referenced by `ng_ubt_connect()`, and `ng_ubt_disconnect()`.

7.28.1.31 USB_ATTACH(ubt)

Definition at line 310 of file `ng_ubt.c`.

References `NG_BT_MBUFQ_INIT`, `NG_UBT_STAT_RESET`, `NG_UBT_WARN_LEVEL`, and `UBT_DEFAULT_QLEN`.

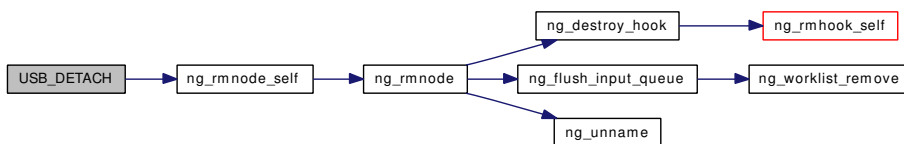
7.28.1.32 USB_DECLARE_DRIVER(ubt)

7.28.1.33 USB_DETACH(ubt)

Definition at line 767 of file `ng_ubt.c`.

References `NG_BT_MBUFQ_DRAIN`, `NG_NODE_SET_PRIVATE`, and `ng_rmnode_self()`.

Here is the call graph for this function:



7.28.1.34 USB_MATCH(ubt)

Definition at line 257 of file `ng_ubt.c`.

7.28.2 Variable Documentation

7.28.2.1 struct `ng_cmdlist ng_ubt_cmdlist[]` [static]

Definition at line 145 of file `ng_ubt.c`.

7.28.2.2 `ng_connect_t ng_ubt_connect` [static]

Definition at line 111 of file ng_ubt.c.

7.28.2.3 `ng_constructor_t ng_ubt_constructor` [static]

Definition at line 108 of file ng_ubt.c.

7.28.2.4 `ng_disconnect_t ng_ubt_disconnect` [static]

Definition at line 112 of file ng_ubt.c.

7.28.2.5 `ng_newhook_t ng_ubt_newhook` [static]

Definition at line 110 of file ng_ubt.c.

7.28.2.6 `struct ng_parse_type ng_ubt_node_qlen_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_ubt_node_qlen_type_fields
}
```

Definition at line 123 of file ng_ubt.c.

7.28.2.7 `struct ng_parse_struct_field ng_ubt_node_qlen_type_fields[]` [static]**Initial value:**

```
{
    { "queue", &ng_parse_int32_type, },
    { "qlen", &ng_parse_int32_type, },
    { NULL, }
}
```

Definition at line 117 of file ng_ubt.c.

7.28.2.8 `struct ng_parse_type ng_ubt_node_stat_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_ubt_node_stat_type_fields
}
```

Definition at line 139 of file ng_ubt.c.

7.28.2.9 `struct ng_parse_struct_field ng_ubt_node_stat_type_fields[]` [static]**Initial value:**

```
{
    { "pkts_recv", &ng_parse_uint32_type, },
    { "bytes_recv", &ng_parse_uint32_type, },
    { "pkts_sent", &ng_parse_uint32_type, },
    { "bytes_sent", &ng_parse_uint32_type, },
    { "oerrors", &ng_parse_uint32_type, },
    { "ierrors", &ng_parse_uint32_type, },
    { NULL, }
}
```

Definition at line 129 of file ng_ubt.c.

7.28.2.10 `ng_rcvdata_t ng_ubt_rcvdata` [static]

Definition at line 114 of file ng_ubt.c.

7.28.2.11 `ng_rcvmsg_t ng_ubt_rcvmsg` [static]

Definition at line 113 of file ng_ubt.c.

7.28.2.12 `ng_shutdown_t ng_ubt_shutdown` [static]

Definition at line 109 of file ng_ubt.c.

7.28.2.13 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_UBT_NODE_TYPE,
    .constructor = ng_ubt_constructor,
    .rcvmsg =      ng_ubt_rcvmsg,
    .shutdown =    ng_ubt_shutdown,
    .newhook =     ng_ubt_newhook,
    .connect =     ng_ubt_connect,
    .rcvdata =     ng_ubt_rcvdata,
    .disconnect =  ng_ubt_disconnect,
    .cmdlist =     ng_ubt_cmdlist
}
```

Definition at line 192 of file ng_ubt.c.

7.29 /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ubt_softc](#)

Defines

- #define [NG_UBT_M_PULLUP](#)(m, s)
- #define [NG_UBT_ALERT](#) if (sc → sc_debug >= NG_UBT_ALERT_LEVEL) printf
- #define [NG_UBT_ERR](#) if (sc → sc_debug >= NG_UBT_ERR_LEVEL) printf
- #define [NG_UBT_WARN](#) if (sc → sc_debug >= NG_UBT_WARN_LEVEL) printf
- #define [NG_UBT_INFO](#) if (sc → sc_debug >= NG_UBT_INFO_LEVEL) printf
- #define [UBT_HCI_REQUEST](#) 0x20
- #define [UBT_DEFAULT_QLEN](#) 12
- #define [UBT_NEED_FRAME_TYPE](#) (1 << 0)
- #define [UBT_HAVE_FRAME_TYPE](#) UBT_NEED_FRAME_TYPE
- #define [UBT_CMD_XMIT](#) (1 << 1)
- #define [UBT_ACL_XMIT](#) (1 << 2)
- #define [UBT_SCO_XMIT](#) (1 << 3)
- #define [UBT_EVT_RECV](#) (1 << 4)
- #define [UBT_ACL_RECV](#) (1 << 5)
- #define [UBT_SCO_RECV](#) (1 << 6)
- #define [UBT_CTRL_DEV](#) (1 << 7)
- #define [UBT_INTR_DEV](#) (1 << 8)
- #define [UBT_BULK_DEV](#) (1 << 9)
- #define [UBT_ANY_DEV](#) (UBT_CTRL_DEV|UBT_INTR_DEV|UBT_BULK_DEV)
- #define [NG_UBT_STAT_PCKTS_SENT](#)(s) (s).pckts_sent ++
- #define [NG_UBT_STAT_BYTES_SENT](#)(s, n) (s).bytes_sent += (n)
- #define [NG_UBT_STAT_PCKTS_RECV](#)(s) (s).pckts_recv ++
- #define [NG_UBT_STAT_BYTES_RECV](#)(s, n) (s).bytes_recv += (n)
- #define [NG_UBT_STAT_OERROR](#)(s) (s).oerrors ++
- #define [NG_UBT_STAT_IERROR](#)(s) (s).ierrors ++
- #define [NG_UBT_STAT_RESET](#)(s) bzero(&(s), sizeof((s)))
- #define [UBT_CTRL_BUFFER_SIZE](#) (sizeof(ng_hci_cmd_pkt_t) + NG_HCI_CMD_PKT_SIZE)
- #define [UBT_BULK_BUFFER_SIZE](#) MCLBYTES
- #define [UBT_ISOC_BUFFER_SIZE](#) (sizeof(ng_hci_scodata_pkt_t) + NG_HCI_SCO_PKT_SIZE)

Typedefs

- typedef [ubt_softc](#) [ubt_softc_t](#)
- typedef [ubt_softc](#) * [ubt_softc_p](#)

7.29.1 Define Documentation

7.29.1.1 #define NG_UBT_ALERT if (sc → sc_debug >= NG_UBT_ALERT_LEVEL) printf

Definition at line 48 of file ng_ubt_var.h.

Referenced by ng_ubt_connect(), and ubt_isoc_in_complete2().

7.29.1.2 #define NG_UBT_ERR if (sc → sc_debug >= NG_UBT_ERR_LEVEL) printf

Definition at line 49 of file ng_ubt_var.h.

Referenced by ng_ubt_rcvdata(), ubt_bulk_in_complete2(), ubt_bulk_in_start(), ubt_bulk_out_start(), ubt_intr_complete2(), ubt_intr_start(), ubt_isoc_in_start(), ubt_isoc_out_start(), ubt_request_complete2(), and ubt_request_start().

7.29.1.3 #define NG_UBT_INFO if (sc → sc_debug >= NG_UBT_INFO_LEVEL) printf

Definition at line 51 of file ng_ubt_var.h.

Referenced by ubt_bulk_in_complete2(), ubt_bulk_out_complete2(), ubt_bulk_out_start(), ubt_intr_complete2(), ubt_isoc_in_complete2(), ubt_isoc_out_complete2(), ubt_isoc_out_start(), ubt_request_complete2(), and ubt_request_start().

7.29.1.4 #define NG_UBT_M_PULLUP(m, s)

Value:

```
do { \
    if ((m)->m_len < (s)) \
        (m) = m_pullup((m), (s)); \
    if ((m) == NULL) \
        NG_UBT_ALERT("%s: %s - m_pullup(%d) failed\n", \
            __func__, device_get_nameunit(sc->sc_dev), (s)); \
} while (0)
```

Definition at line 38 of file ng_ubt_var.h.

7.29.1.5 #define NG_UBT_STAT_BYTES_RECV(s, n) (s).bytes_recv += (n)

Definition at line 79 of file ng_ubt_var.h.

Referenced by ubt_bulk_in_complete2(), ubt_intr_complete2(), and ubt_isoc_in_complete2().

7.29.1.6 #define NG_UBT_STAT_BYTES_SENT(s, n) (s).bytes_sent += (n)

Definition at line 77 of file ng_ubt_var.h.

Referenced by ubt_bulk_out_complete2(), ubt_isoc_out_complete2(), and ubt_request_complete2().

7.29.1.7 #define NG_UBT_STAT_IERROR(s) (s).ierrors ++

Definition at line 81 of file ng_ubt_var.h.

Referenced by `ubt_bulk_in_complete2()`, `ubt_intr_complete2()`, and `ubt_isoc_in_complete2()`.

7.29.1.8 **#define NG_UBT_STAT_OERROR(s) (s).oerrors ++**

Definition at line 80 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_out_complete2()`, `ubt_bulk_out_start()`, `ubt_isoc_out_complete2()`, `ubt_isoc_out_start()`, `ubt_request_complete2()`, and `ubt_request_start()`.

7.29.1.9 **#define NG_UBT_STAT_PCKTS_RECV(s) (s).pckts_recv ++**

Definition at line 78 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_complete2()`, `ubt_intr_complete2()`, and `ubt_isoc_in_complete2()`.

7.29.1.10 **#define NG_UBT_STAT_PCKTS_SENT(s) (s).pckts_sent ++**

Definition at line 76 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_out_complete2()`, `ubt_isoc_out_complete2()`, and `ubt_request_complete2()`.

7.29.1.11 **#define NG_UBT_STAT_RESET(s) bzero(&(s), sizeof((s)))**

Definition at line 82 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvmsg()`, and `USB_ATTACH()`.

7.29.1.12 **#define NG_UBT_WARN if (sc → sc_debug >= NG_UBT_WARN_LEVEL) printf**

Definition at line 50 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_complete2()`, `ubt_bulk_out_complete2()`, `ubt_intr_complete2()`, `ubt_isoc_in_complete2()`, and `ubt_isoc_out_complete2()`.

7.29.1.13 **#define UBT_ACL_RECV (1 << 5)**

Definition at line 68 of file `ng_ubt_var.h`.

Referenced by `ubt_bulk_in_complete2()`, and `ubt_bulk_in_start()`.

7.29.1.14 **#define UBT_ACL_XMIT (1 << 2)**

Definition at line 65 of file `ng_ubt_var.h`.

Referenced by `ng_ubt_rcvdata()`, `ubt_bulk_out_complete2()`, and `ubt_bulk_out_start()`.

7.29.1.15 **#define UBT_ANY_DEV (UBT_CTRL_DEV|UBT_INTR_DEV|UBT_BULK_DEV)**

Definition at line 73 of file `ng_ubt_var.h`.

7.29.1.16 #define UBT_BULK_BUFFER_SIZE MCLBYTES

Definition at line 116 of file ng_ubt_var.h.

Referenced by ubt_bulk_out_start().

7.29.1.17 #define UBT_BULK_DEV (1 << 9)

Definition at line 72 of file ng_ubt_var.h.

7.29.1.18 #define UBT_CMD_XMIT (1 << 1)

Definition at line 64 of file ng_ubt_var.h.

Referenced by ng_ubt_rcvdata(), ubt_request_complete2(), and ubt_request_start().

7.29.1.19 #define UBT_CTRL_BUFFER_SIZE (sizeof(ng_hci_cmd_pkt_t) + NG_HCI_CMD_PKT_SIZE)

Definition at line 101 of file ng_ubt_var.h.

Referenced by ubt_request_start().

7.29.1.20 #define UBT_CTRL_DEV (1 << 7)

Definition at line 70 of file ng_ubt_var.h.

7.29.1.21 #define UBT_DEFAULT_QLEN 12

Definition at line 55 of file ng_ubt_var.h.

Referenced by USB_ATTACH().

7.29.1.22 #define UBT_EVT_RECV (1 << 4)

Definition at line 67 of file ng_ubt_var.h.

Referenced by ubt_intr_complete2(), and ubt_intr_start().

7.29.1.23 #define UBT_HAVE_FRAME_TYPE UBT_NEED_FRAME_TYPE

Definition at line 63 of file ng_ubt_var.h.

Referenced by ubt_bulk_in_start(), ubt_intr_start(), and ubt_isoc_in_complete2().

7.29.1.24 #define UBT_HCI_REQUEST 0x20

Definition at line 54 of file ng_ubt_var.h.

Referenced by ubt_request_start().

7.29.1.25 #define UBT_INTR_DEV (1 << 8)

Definition at line 71 of file ng_ubt_var.h.

7.29.1.26 #define UBT_ISOC_BUFFER_SIZE (sizeof(ng_hci_scodata_pkt_t) + NG_HCI_SCO_PKT_SIZE)

Definition at line 136 of file ng_ubt_var.h.

7.29.1.27 #define UBT_NEED_FRAME_TYPE (1 << 0)

Definition at line 62 of file ng_ubt_var.h.

Referenced by ng_ubt_rcvdata().

7.29.1.28 #define UBT_SCO_RECV (1 << 6)

Definition at line 69 of file ng_ubt_var.h.

Referenced by ubt_isoc_in_complete2(), and ubt_isoc_in_start().

7.29.1.29 #define UBT_SCO_XMIT (1 << 3)

Definition at line 66 of file ng_ubt_var.h.

Referenced by ng_ubt_rcvdata(), ubt_isoc_out_complete2(), and ubt_isoc_out_start().

7.29.2 Typedef Documentation**7.29.2.1 typedef struct [ubt_softc*](#) [ubt_softc_p](#)**

Definition at line 144 of file ng_ubt_var.h.

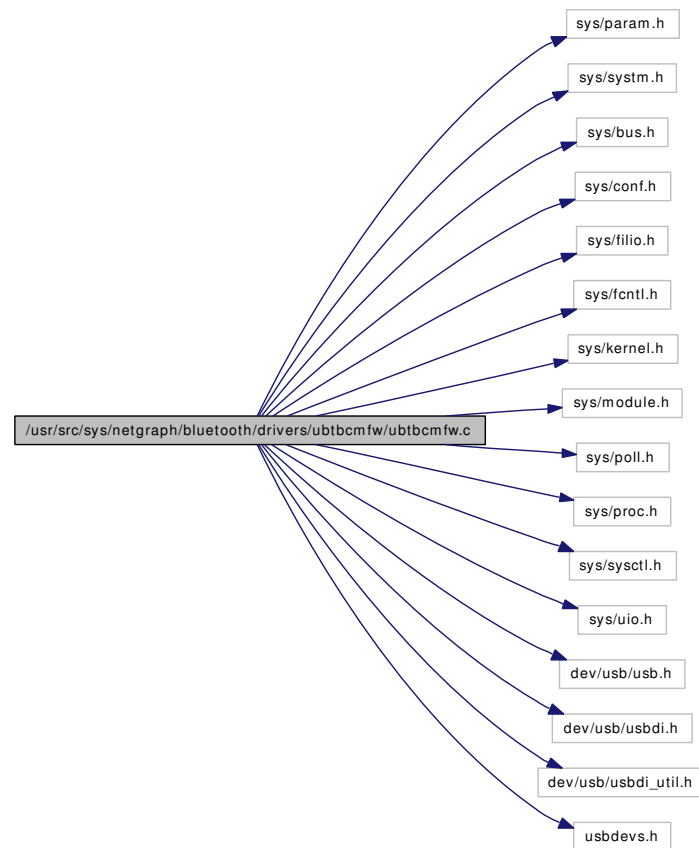
7.29.2.2 typedef struct [ubt_softc](#) [ubt_softc_t](#)

Definition at line 143 of file ng_ubt_var.h.

7.30 /usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ubtbcmfw.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bus.h>
#include <sys/conf.h>
#include <sys/filio.h>
#include <sys/fcntl.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/poll.h>
#include <sys/proc.h>
#include <sys/sysctl.h>
#include <sys/uio.h>
#include <dev/usb/usb.h>
#include <dev/usb/usbdi.h>
#include <dev/usb/usbdi_util.h>
#include "usbdevs.h"
```

Include dependency graph for ubtbcmfw.c:



Data Structures

- struct `ubtbcmfw_softc`

Defines

- `#define USBBASEDEVICE` `device_t`
- `#define USBDEVNAME` `device_get_nameunit`
- `#define USBDEVUNIT(bdev)` `device_get_unit(bdev)`
- `#define UBTBCMFW_CONFIG_NO` `1`
- `#define UBTBCMFW_IFACE_IDX` `0`
- `#define UBTBCMFW_INTR_IN_EP` `0x81`
- `#define UBTBCMFW_BULK_OUT_EP` `0x02`
- `#define UBTBCMFW_INTR_IN` `UE_GET_ADDR(UBTBCMFW_INTR_IN_EP)`
- `#define UBTBCMFW_BULK_OUT` `UE_GET_ADDR(UBTBCMFW_BULK_OUT_EP)`
- `#define UBTBCMFW_CTRL_DEV` `(1 << 0)`
- `#define UBTBCMFW_INTR_IN_DEV` `(1 << 1)`
- `#define UBTBCMFW_BULK_OUT_DEV` `(1 << 2)`
- `#define UBTBCMFW_UNIT(n)` `((minor(n) >> 4) & 0xf)`
- `#define UBTBCMFW_ENDPOINT(n)` `(minor(n) & 0xf)`
- `#define UBTBCMFW_MINOR(u, e)` `((u << 4) | (e))`
- `#define UBTBCMFW_BSIZE` `1024`
- `#define USB_PRODUCT_BROADCOM_BCM2033NF` `0x2033`

Typedefs

- typedef `ubtbcmfw_softc * ubtbcmfw_softc_p`

Functions

- `USB_DECLARE_DRIVER` (ubtbcmfw)
- `DRIVER_MODULE` (ubtbcmfw, uhub, ubtbcmfw_driver, ubtbcmfw_devclass, usbd_driver_load, 0)
- `USB_MATCH` (ubtbcmfw)
- `USB_ATTACH` (ubtbcmfw)
- `USB_DETACH` (ubtbcmfw)
- static int `ubtbcmfw_open` (struct cdev *dev, int flag, int mode, usb_proc_ptr p)
- static int `ubtbcmfw_close` (struct cdev *dev, int flag, int mode, usb_proc_ptr p)
- static int `ubtbcmfw_read` (struct cdev *dev, struct uio *uio, int flag)
- static int `ubtbcmfw_write` (struct cdev *dev, struct uio *uio, int flag)
- static int `ubtbcmfw_ioctl` (struct cdev *dev, u_long cmd, caddr_t data, int flag, usb_proc_ptr p)
- static int `ubtbcmfw_poll` (struct cdev *dev, int events, usb_proc_ptr p)

Variables

- static d_open_t `ubtbcmfw_open`
- static d_close_t `ubtbcmfw_close`
- static d_read_t `ubtbcmfw_read`
- static d_write_t `ubtbcmfw_write`
- static d_ioctl_t `ubtbcmfw_ioctl`
- static d_poll_t `ubtbcmfw_poll`
- static struct cdevsw `ubtbcmfw_cdevsw`

7.30.1 Define Documentation

7.30.1.1 #define UBTBCMFW_BSIZE 1024

Definition at line 95 of file ubtbcmfw.c.

Referenced by `ubtbcmfw_read()`, and `ubtbcmfw_write()`.

7.30.1.2 #define UBTBCMFW_BULK_OUT UE_GET_ADDR(UBTBCMFW_BULK_OUT_EP)

Definition at line 68 of file ubtbcmfw.c.

Referenced by `ubtbcmfw_close()`, `ubtbcmfw_open()`, `ubtbcmfw_poll()`, `ubtbcmfw_write()`, and `USB_ATTACH()`.

7.30.1.3 #define UBTBCMFW_BULK_OUT_DEV (1 << 2)

Definition at line 81 of file ubtbcmfw.c.

Referenced by `ubtbcmfw_close()`, and `ubtbcmfw_open()`.

7.30.1.4 #define UBTBCMFW_BULK_OUT_EP 0x02

Definition at line 66 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.5 #define UBTBCMFW_CONFIG_NO 1

Definition at line 63 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.6 #define UBTBCMFW_CTRL_DEV (1 << 0)

Definition at line 79 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), and ubtbcmfw_open().

7.30.1.7 #define UBTBCMFW_ENDPOINT(n) (minor(n) & 0xf)

Definition at line 93 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), ubtbcmfw_ioctl(), ubtbcmfw_open(), ubtbcmfw_poll(), ubtbcmfw_read(), and ubtbcmfw_write().

7.30.1.8 #define UBTBCMFW_IFACE_IDX 0

Definition at line 64 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.9 #define UBTBCMFW_INTR_IN UE_GET_ADDR(UBTBCMFW_INTR_IN_EP)

Definition at line 67 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), ubtbcmfw_open(), ubtbcmfw_poll(), ubtbcmfw_read(), and USB_ATTACH().

7.30.1.10 #define UBTBCMFW_INTR_IN_DEV (1 << 1)

Definition at line 80 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), and ubtbcmfw_open().

7.30.1.11 #define UBTBCMFW_INTR_IN_EP 0x81

Definition at line 65 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.12 #define UBTBCMFW_MINOR(u, e) (((u) << 4) | (e))

Definition at line 94 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.13 #define UBTBCMFW_UNIT(n) ((minor(n) >> 4) & 0xf)

Definition at line 92 of file ubtbcmfw.c.

Referenced by ubtbcmfw_close(), ubtbcmfw_ioctl(), ubtbcmfw_open(), ubtbcmfw_poll(), ubtbcmfw_read(), and ubtbcmfw_write().

7.30.1.14 #define USB_PRODUCT_BROADCOM_BCM2033NF 0x2033

Referenced by USB_MATCH().

7.30.1.15 #define USBBASEDEVICE device_t

Definition at line 55 of file ubtbcmfw.c.

7.30.1.16 #define USBDEVNAME device_get_nameunit

Definition at line 56 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.1.17 #define USBDEVUNIT(bdev) device_get_unit(bdev)

Definition at line 57 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.2 Typedef Documentation**7.30.2.1 typedef struct [ubtbcmfw_softc*](#) [ubtbcmfw_softc_p](#)**

Definition at line 86 of file ubtbcmfw.c.

7.30.3 Function Documentation**7.30.3.1 DRIVER_MODULE (ubtbcmfw, uhub, ubtbcmfw_driver, ubtbcmfw_devclass, usbd_driver_load, 0)****7.30.3.2 static int ubtbcmfw_close (struct cdev * dev, int flag, int mode, usb_proc_ptr p) [static]**

Definition at line 328 of file ubtbcmfw.c.

References `ubtbcmfw_softc::sc_bulk_out_pipe`, `ubtbcmfw_softc::sc_flags`, `ubtbcmfw_softc::sc_intr_in_pipe`, `UBTBCMFW_BULK_OUT`, `UBTBCMFW_BULK_OUT_DEV`, `UBTBCMFW_CTRL_DEV`, `UBTBCMFW_ENDPOINT`, `UBTBCMFW_INTR_IN`, `UBTBCMFW_INTR_IN_DEV`, and `UBTBCMFW_UNIT`.

7.30.3.3 `static int ubtbcmfw_ioctl (struct cdev * dev, u_long cmd, caddr_t data, int flag, usb_proc_ptr p)` [static]

Definition at line 494 of file `ubtbcmfw.c`.

References `ubtbcmfw_softc::sc_dev`, `ubtbcmfw_softc::sc_dying`, `ubtbcmfw_softc::sc_refcnt`, `ubtbcmfw_softc::sc_udev`, `UBTBCMFW_ENDPOINT`, and `UBTBCMFW_UNIT`.

7.30.3.4 `static int ubtbcmfw_open (struct cdev * dev, int flag, int mode, usb_proc_ptr p)` [static]

Definition at line 276 of file `ubtbcmfw.c`.

References `ubtbcmfw_softc::sc_bulk_out_pipe`, `ubtbcmfw_softc::sc_dying`, `ubtbcmfw_softc::sc_flags`, `ubtbcmfw_softc::sc_intr_in_pipe`, `UBTBCMFW_BULK_OUT`, `UBTBCMFW_BULK_OUT_DEV`, `UBTBCMFW_CTRL_DEV`, `UBTBCMFW_ENDPOINT`, `UBTBCMFW_INTR_IN`, `UBTBCMFW_INTR_IN_DEV`, and `UBTBCMFW_UNIT`.

7.30.3.5 `static int ubtbcmfw_poll (struct cdev * dev, int events, usb_proc_ptr p)` [static]

Definition at line 531 of file `ubtbcmfw.c`.

References `ubtbcmfw_softc::sc_bulk_out_pipe`, `ubtbcmfw_softc::sc_intr_in_pipe`, `UBTBCMFW_BULK_OUT`, `UBTBCMFW_ENDPOINT`, `UBTBCMFW_INTR_IN`, and `UBTBCMFW_UNIT`.

7.30.3.6 `static int ubtbcmfw_read (struct cdev * dev, struct uio * uio, int flag)` [static]

Definition at line 365 of file `ubtbcmfw.c`.

References `min`, `ubtbcmfw_softc::sc_dev`, `ubtbcmfw_softc::sc_dying`, `ubtbcmfw_softc::sc_intr_in_pipe`, `ubtbcmfw_softc::sc_refcnt`, `ubtbcmfw_softc::sc_udev`, `UBTBCMFW_BSIZE`, `UBTBCMFW_ENDPOINT`, `UBTBCMFW_INTR_IN`, and `UBTBCMFW_UNIT`.

7.30.3.7 `static int ubtbcmfw_write (struct cdev * dev, struct uio * uio, int flag)` [static]

Definition at line 429 of file `ubtbcmfw.c`.

References `min`, `ubtbcmfw_softc::sc_bulk_out_pipe`, `ubtbcmfw_softc::sc_dev`, `ubtbcmfw_softc::sc_dying`, `ubtbcmfw_softc::sc_refcnt`, `ubtbcmfw_softc::sc_udev`, `UBTBCMFW_BSIZE`, `UBTBCMFW_BULK_OUT`, `UBTBCMFW_ENDPOINT`, and `UBTBCMFW_UNIT`.

7.30.3.8 `USB_ATTACH (ubtbcmfw)`

Definition at line 149 of file `ubtbcmfw.c`.

References `UBTBCMFW_BULK_OUT`, `UBTBCMFW_BULK_OUT_EP`, `ubtbcmfw_cdevsw`, `UBTBCMFW_CONFIG_NO`, `UBTBCMFW_IFACE_IDX`, `UBTBCMFW_INTR_IN`, `UBTBCMFW_INTR_IN_EP`, `UBTBCMFW_MINOR`, `USBDEVNAME`, and `USBDEVUNIT`.

7.30.3.9 USB_DECLARE_DRIVER (ubtbcmfw)**7.30.3.10 USB_DETACH (ubtbcmfw)**

Definition at line 224 of file ubtbcmfw.c.

7.30.3.11 USB_MATCH (ubtbcmfw)

Definition at line 128 of file ubtbcmfw.c.

References USB_PRODUCT_BROADCOM_BCM2033NF.

7.30.4 Variable Documentation**7.30.4.1 struct cdevsw ubtbcmfw_cdevsw** [static]

Initial value:

```
{
    .d_version =    D_VERSION,
    .d_flags =     D_NEEDGIANT,
    .d_open =      ubtbcmfw_open,
    .d_close =     ubtbcmfw_close,
    .d_read =      ubtbcmfw_read,
    .d_write =     ubtbcmfw_write,
    .d_ioctl =     ubtbcmfw_ioctl,
    .d_poll =      ubtbcmfw_poll,
    .d_name =      "ubtbcmfw",
}
```

Definition at line 104 of file ubtbcmfw.c.

Referenced by USB_ATTACH().

7.30.4.2 d_close_t ubtbcmfw_close [static]

Definition at line 98 of file ubtbcmfw.c.

7.30.4.3 d_ioctl_t ubtbcmfw_ioctl [static]

Definition at line 101 of file ubtbcmfw.c.

7.30.4.4 d_open_t ubtbcmfw_open [static]

Definition at line 97 of file ubtbcmfw.c.

7.30.4.5 d_poll_t ubtbcmfw_poll [static]

Definition at line 102 of file ubtbcmfw.c.

7.30.4.6 d_read_t ubtbcmfw_read [static]

Definition at line 99 of file ubtbcmfw.c.

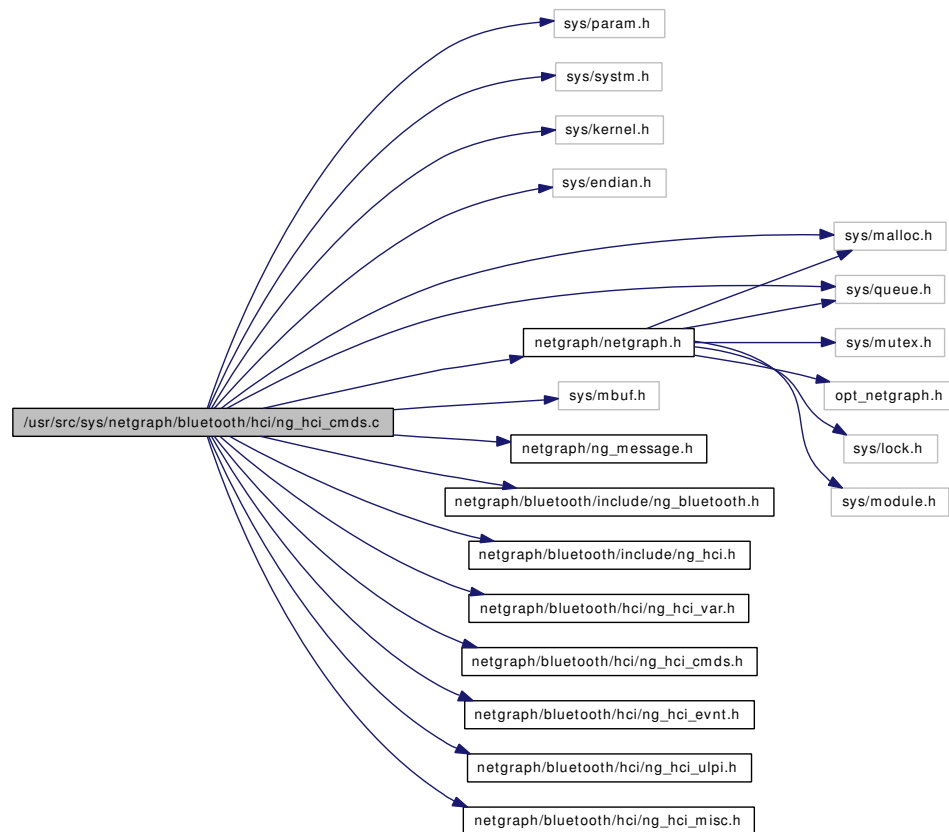
7.30.4.7 d_write_t ubtbcmfw_write [static]

Definition at line 100 of file ubtbcmfw.c.

7.31 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/hci/ng_hci_var.h>
#include <netgraph/bluetooth/hci/ng_hci_cmds.h>
#include <netgraph/bluetooth/hci/ng_hci_evnt.h>
#include <netgraph/bluetooth/hci/ng_hci_ulpi.h>
#include <netgraph/bluetooth/hci/ng_hci_misc.h>
```

Include dependency graph for ng_hci_cmds.c:



Defines

- #define `min(a, b)` `((a) < (b)) ? (a) : (b)`

Functions

- static int `complete_command` (`ng_hci_unit_p`, int, struct mbuf **)
- static int `process_link_control_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_link_policy_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_hc_baseband_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_info_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_status_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_testing_params` (`ng_hci_unit_p`, u_int16_t, struct mbuf *, struct mbuf *)
- static int `process_link_control_status` (`ng_hci_unit_p`, ng_hci_command_status_ep *, struct mbuf *)
- static int `process_link_policy_status` (`ng_hci_unit_p`, ng_hci_command_status_ep *, struct mbuf *)
- int `ng_hci_send_command` (`ng_hci_unit_p` unit)
- int `ng_hci_process_command_complete` (`ng_hci_unit_p` unit, struct mbuf *e)
- int `ng_hci_process_command_status` (`ng_hci_unit_p` unit, struct mbuf *e)
- void `ng_hci_process_command_timeout` (`node_p` node, `hook_p` hook, void *arg1, int arg2)

7.31.1 Define Documentation

7.31.1.1 #define min(a, b) ((a) < (b)) ? (a) : (b)

Definition at line 58 of file ng_hci_cmds.c.

Referenced by bt3c_send(), ng_btsocket_hci_raw_control(), ng_btsocket_l2cap_raw_control(), ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_prepare_packet(), ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_session_accept(), ng_btsocket_rfcomm_session_connect(), ng_h4_input(), process_info_params(), send_acname(), ubt_isoc_in_complete2(), ubt_isoc_out_start(), ubtbcmfw_read(), and ubtbcmfw_write().

7.31.2 Function Documentation

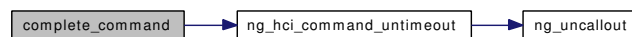
7.31.2.1 static int complete_command (ng_hci_unit_p, int, struct mbuf **) [static]

Definition at line 324 of file ng_hci_cmds.c.

References ng_hci_unit::cmdq, NG_BT_MBUFQ_DEQUEUE, NG_BT_MBUFQ_FIRST, NG_HCI_ALERT, ng_hci_command_untimeout(), NG_HCI_UNIT_COMMAND_PENDING, NG_NODE_NAME, ng_hci_unit::node, and ng_hci_unit::state.

Referenced by ng_hci_process_command_complete(), and ng_hci_process_command_status().

Here is the call graph for this function:



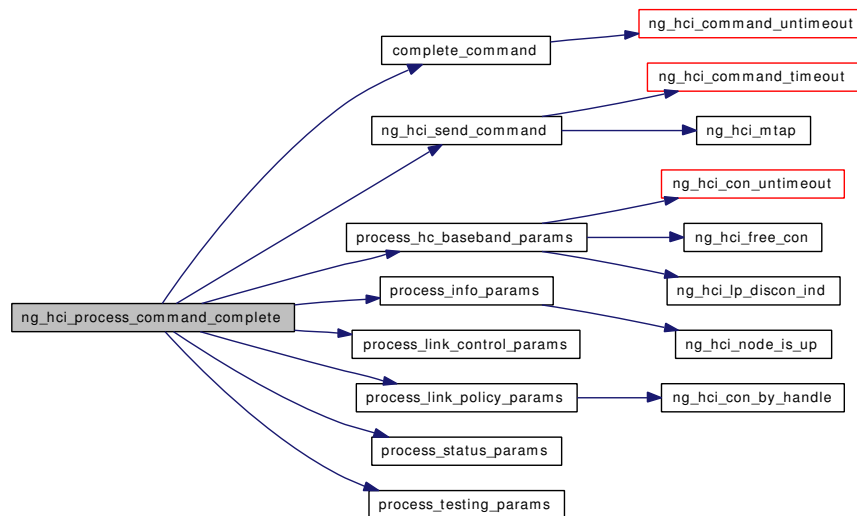
7.31.2.2 int ng_hci_process_command_complete (ng_hci_unit_p unit, struct mbuf * e)

Definition at line 158 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, complete_command(), NG_FREE_M, NG_HCI_BUFF_CMD_SET, NG_HCI_ERR, NG_HCI_M_PULLUP, NG_HCI_OCF, NG_HCI_OGF, NG_HCI_OGF_BT_LOGO, NG_HCI_OGF_HC_BASEBAND, NG_HCI_OGF_INFO, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OGF_LINK_POLICY, NG_HCI_OGF_STATUS, NG_HCI_OGF_TESTING, NG_HCI_OGF_VENDOR, ng_hci_send_command(), NG_NODE_NAME, ng_hci_unit::node, process_hc_baseband_params(), process_info_params(), process_link_control_params(), process_link_policy_params(), process_status_params(), and process_testing_params().

Referenced by ng_hci_process_event().

Here is the call graph for this function:



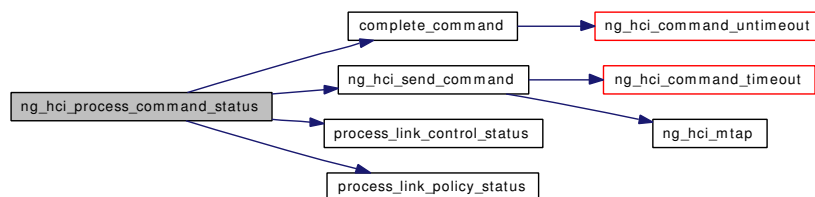
7.31.2.3 int ng_hci_process_command_status (ng_hci_unit_p unit, struct mbuf * e)

Definition at line 260 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, complete_command(), NG_FREE_M, NG_HCI_BUFF_CMD_SET, NG_HCI_M_PULLUP, NG_HCI_OGF, NG_HCI_OGF_BT_LOGO, NG_HCI_OGF_HC_BASEBAND, NG_HCI_OGF_INFO, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OGF_LINK_POLICY, NG_HCI_OGF_STATUS, NG_HCI_OGF_TESTING, NG_HCI_OGF_VENDOR, ng_hci_send_command(), process_link_control_status(), and process_link_policy_status().

Referenced by ng_hci_process_event().

Here is the call graph for this function:



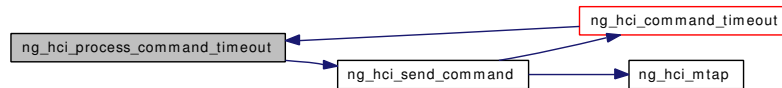
7.31.2.4 void ng_hci_process_command_timeout (node_p node, hook_p hook, void * arg1, int arg2)

Definition at line 382 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, ng_hci_unit::cmdq, NG_BT_MBUFQ_DEQUEUE, NG_FREE_M, NG_HCI_ALERT, NG_HCI_BUFF_CMD_SET, NG_HCI_ERR, NG_HCI_OCF, NG_HCI_OGF, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, ng_hci_unit::node, and ng_hci_unit::state.

Referenced by ng_hci_command_timeout().

Here is the call graph for this function:



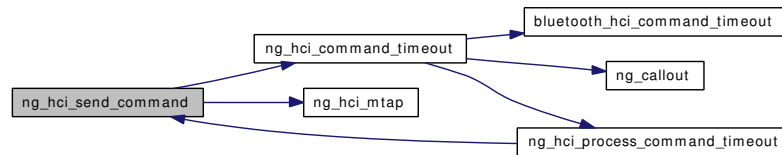
7.31.2.5 int ng_hci_send_command (ng_hci_unit_p unit)

Definition at line 85 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, ng_hci_unit::cmdq, ng_hci_unit::drv, NG_BT_MBUFQ_DRAIN, NG_BT_MBUFQ_FIRST, NG_HCI_BUFF_CMD_GET, NG_HCI_BUFF_CMD_USE, ng_hci_command_timeout(), NG_HCI_ERR, NG_HCI_HOOK_DRV, ng_hci_mtap(), NG_HCI_STAT_BYTES_SENT, NG_HCI_STAT_CMD_SENT, NG_HCI_UNIT_COMMAND_PENDING, NG_HCI_WARN, NG_HOOK_NOT_VALID, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_hci_unit::node, ng_hci_unit::stat, and ng_hci_unit::state.

Referenced by con_compl(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_process_command_complete(), ng_hci_process_command_status(), ng_hci_process_command_timeout(), and ng_hci_raw_rcvdata().

Here is the call graph for this function:



7.31.2.6 int process_hc_baseband_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *) [static]

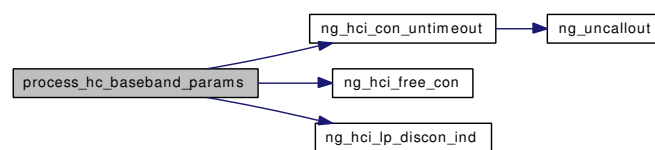
Definition at line 551 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, ng_hci_unit_con::flags, NG_FREE_M, NG_HCI_BUFF_ACL_FREE, NG_HCI_BUFF_ACL_TOTAL, NG_HCI_BUFF_SCO_FREE, NG_HCI_BUFF_SCO_TOTAL, NG_HCI_CON_TIMEOUT_PENDING, ng_hci_con_untimeout(), ng_hci_free_con(), ng_hci_lp_discon_ind(), NG_HCI_OCF_CHANGE_LOCAL_NAME, NG_HCI_OCF_CREATE_NEW_UNIT_KEY, NG_HCI_OCF_DELETE_STORED_LINK_KEY, NG_HCI_OCF_FLUSH, NG_HCI_OCF_H2HC_FLOW_CONTROL, NG_HCI_OCF_HOST_BUFFER_SIZE, NG_HCI_OCF_HOST_NUM_COMPL_PKTS, NG_HCI_OCF_READ_AUTH_ENABLE, NG_HCI_OCF_READ_AUTO_FLUSH_TIMO, NG_HCI_OCF_READ_CON_ACCEPT_TIMO, NG_HCI_OCF_READ_ENCRYPTION_MODE, NG_HCI_OCF_READ_HOLD_MODE_ACTIVITY, NG_HCI_OCF_READ_IAC_LAP, NG_HCI_OCF_READ_INQUIRY_SCAN_ACTIVITY, NG_HCI_OCF_READ_LINK_SUPERVISION_TIMO, NG_HCI_OCF_READ_LOCAL_NAME, NG_HCI_OCF_READ_NUM_BROADCAST_RETRANS, NG_HCI_OCF_READ_PAGE_SCAN, NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY, NG_HCI_OCF_READ_PAGE_SCAN_PERIOD, NG_HCI_OCF_READ_PAGE_TIMO, NG_HCI_OCF_READ_PIN_TYPE, NG_HCI_OCF_READ_SCAN_ENABLE, NG_HCI_OCF_READ_SCO_FLOW_CONTROL, NG_HCI_OCF_READ_STORED_LINK_KEY, NG_HCI_OCF_READ_SUPPORTED_IAC_NUM, NG_HCI_OCF_READ_UNIT_CLASS, NG_HCI_OCF_READ_VOICE_SETTINGS, NG_HCI_OCF_READ_XMIT_LEVEL, NG_HCI_OCF_RESET, NG_HCI_OCF_SET_EVENT_FILTER,

NG_HCI_OCF_SET_EVENT_MASK, NG_HCI_OCF_WRITE_AUTH_ENABLE, NG_HCI_OCF_WRITE_AUTO_FLUSH_TIMO, NG_HCI_OCF_WRITE_CON_ACCEPT_TIMO, NG_HCI_OCF_WRITE_ENCRYPTION_MODE, NG_HCI_OCF_WRITE_HOLD_MODE_ACTIVITY, NG_HCI_OCF_WRITE_IAC_LAP, NG_HCI_OCF_WRITE_INQUIRY_SCAN_ACTIVITY, NG_HCI_OCF_WRITE_LINK_SUPERVISION_TIMO, NG_HCI_OCF_WRITE_NUM_BROADCAST_RETRANS, NG_HCI_OCF_WRITE_PAGE_SCAN, NG_HCI_OCF_WRITE_PAGE_SCAN_ACTIVITY, NG_HCI_OCF_WRITE_PAGE_SCAN_PERIOD, NG_HCI_OCF_WRITE_PAGE_TIMO, NG_HCI_OCF_WRITE_PIN_TYPE, NG_HCI_OCF_WRITE_SCAN_ENABLE, NG_HCI_OCF_WRITE_SCO_FLOW_CONTROL, NG_HCI_OCF_WRITE_STORED_LINK_KEY, NG_HCI_OCF_WRITE_UNIT_CLASS, NG_HCI_OCF_WRITE_VOICE_SETTINGS, NG_HCI_UNIT_INITED, and ng_hci_unit::state.

Referenced by ng_hci_process_command_complete().

Here is the call graph for this function:



7.31.2.7 static int process_info_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *) [static]

Definition at line 662 of file ng_hci_cmds.c.

References ng_hci_unit::acl, ng_hci_unit::bdaddr, ng_hci_unit::buffer, ng_hci_unit::features, min, NG_FREE_M, NG_HCI_BUFF_ACL_SET, NG_HCI_BUFF_SCO_SET, NG_HCI_M_PULLUP, ng_hci_node_is_up(), NG_HCI_OCF_READ_BDADDR, NG_HCI_OCF_READ_BUFFER_SIZE, NG_HCI_OCF_READ_COUNTRY_CODE, NG_HCI_OCF_READ_LOCAL_FEATURES, NG_HCI_OCF_READ_LOCAL_VER, NG_HCI_UNIT_READY, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit::state.

Referenced by ng_hci_process_command_complete().

Here is the call graph for this function:



7.31.2.8 static int process_link_control_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *) [static]

Definition at line 427 of file ng_hci_cmds.c.

References NG_FREE_M, NG_HCI_OCF_ACCEPT_CON, NG_HCI_OCF_ADD_SCO_CON, NG_HCI_OCF_AUTH_REQ, NG_HCI_OCF_CHANGE_CON_LINK_KEY, NG_HCI_OCF_CHANGE_CON_PKT_TYPE, NG_HCI_OCF_CREATE_CON, NG_HCI_OCF_DISCON, NG_HCI_OCF_EXIT_PERIODIC_INQUIRY, NG_HCI_OCF_INQUIRY, NG_HCI_OCF_INQUIRY_CANCEL, NG_HCI_OCF_LINK_KEY_NEG_REP, NG_HCI_OCF_LINK_KEY_REP, NG_HCI_OCF_MASTER_LINK_KEY, NG_HCI_OCF_PERIODIC_INQUIRY, NG_HCI_OCF_PIN_CODE_NEG_REP, NG_HCI_OCF_PIN_CODE_REP, NG_HCI_OCF_READ_CLOCK_OFFSET, NG_HCI_OCF_READ_REMOTE_

FEATURES, NG_HCI_OCF_READ_REMOTE_VER_INFO, NG_HCI_OCF_REJECT_CON, NG_HCI_OCF_REMOTE_NAME_REQ, and NG_HCI_OCF_SET_CON_ENCRYPTION.

Referenced by `ng_hci_process_command_complete()`.

7.31.2.9 `static int process_link_control_status (ng_hci_unit_p, ng_hci_command_status_ep *, struct mbuf *)` [static]

Definition at line 804 of file `ng_hci_cmds.c`.

References NG_FREE_M, NG_HCI_OCF, NG_HCI_OCF_ACCEPT_CON, NG_HCI_OCF_ADD_SCO_CON, NG_HCI_OCF_AUTH_REQ, NG_HCI_OCF_CHANGE_CON_LINK_KEY, NG_HCI_OCF_CHANGE_CON_PKT_TYPE, NG_HCI_OCF_CREATE_CON, NG_HCI_OCF_DISCON, NG_HCI_OCF_EXIT_PERIODIC_INQUIRY, NG_HCI_OCF_INQUIRY, NG_HCI_OCF_INQUIRY_CANCEL, NG_HCI_OCF_LINK_KEY_NEG_REP, NG_HCI_OCF_LINK_KEY_REP, NG_HCI_OCF_MASTER_LINK_KEY, NG_HCI_OCF_PERIODIC_INQUIRY, NG_HCI_OCF_PIN_CODE_NEG_REP, NG_HCI_OCF_PIN_CODE_REP, NG_HCI_OCF_READ_CLOCK_OFFSET, NG_HCI_OCF_READ_REMOTE_FEATURES, NG_HCI_OCF_READ_REMOTE_VER_INFO, NG_HCI_OCF_REJECT_CON, NG_HCI_OCF_REMOTE_NAME_REQ, and NG_HCI_OCF_SET_CON_ENCRYPTION.

Referenced by `ng_hci_process_command_status()`.

7.31.2.10 `static int process_link_policy_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *)` [static]

Definition at line 482 of file `ng_hci_cmds.c`.

References `ng_hci_unit_con::link_type`, NG_FREE_M, NG_HCI_ALERT, `ng_hci_con_by_handle()`, NG_HCI_CON_HANDLE, NG_HCI_LINK_ACL, NG_HCI_M_PULLUP, NG_HCI_OCF_EXIT_PARK_MODE, NG_HCI_OCF_EXIT_SNIFF_MODE, NG_HCI_OCF_HOLD_MODE, NG_HCI_OCF_PARK_MODE, NG_HCI_OCF_QOS_SETUP, NG_HCI_OCF_READ_LINK_POLICY_SETTINGS, NG_HCI_OCF_ROLE_DISCOVERY, NG_HCI_OCF_SNIFF_MODE, NG_HCI_OCF_SWITCH_ROLE, NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS, NG_NODE_NAME, `ng_hci_unit::node`, and `ng_hci_unit_con::role`.

Referenced by `ng_hci_process_command_complete()`.

Here is the call graph for this function:



7.31.2.11 `static int process_link_policy_status (ng_hci_unit_p, ng_hci_command_status_ep *, struct mbuf *)` [static]

Definition at line 863 of file `ng_hci_cmds.c`.

References NG_FREE_M, NG_HCI_OCF, NG_HCI_OCF_EXIT_PARK_MODE, NG_HCI_OCF_EXIT_SNIFF_MODE, NG_HCI_OCF_HOLD_MODE, NG_HCI_OCF_PARK_MODE, NG_HCI_OCF_QOS_SETUP, NG_HCI_OCF_READ_LINK_POLICY_SETTINGS, NG_HCI_OCF_ROLE_DISCOVERY, NG_HCI_OCF_SNIFF_MODE, NG_HCI_OCF_SWITCH_ROLE, and NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS.

Referenced by `ng_hci_process_command_status()`.

7.31.2.12 `static int process_status_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *)` [static]

Definition at line 740 of file ng_hci_cmds.c.

References NG_FREE_M, NG_HCI_OCF_GET_LINK_QUALITY, NG_HCI_OCF_READ_FAILED_CONTACT_CNTR, NG_HCI_OCF_READ_RSSI, and NG_HCI_OCF_RESET_FAILED_CONTACT_CNTR.

Referenced by ng_hci_process_command_complete().

7.31.2.13 `int process_testing_params (ng_hci_unit_p, u_int16_t, struct mbuf *, struct mbuf *)` [static]

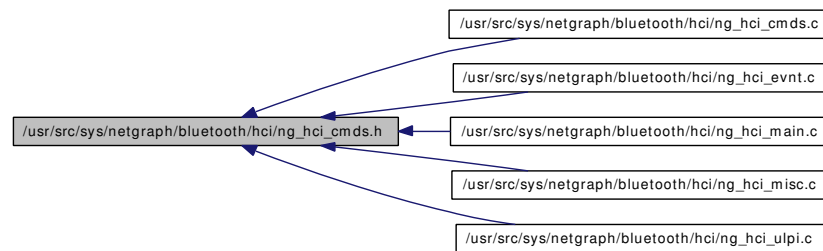
Definition at line 769 of file ng_hci_cmds.c.

References NG_FREE_M, NG_HCI_OCF_ENABLE_UNIT_UNDER_TEST, NG_HCI_OCF_READ_LOOPBACK_MODE, and NG_HCI_OCF_WRITE_LOOPBACK_MODE.

Referenced by ng_hci_process_command_complete().

7.32 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [ng_hci_send_command](#) (ng_hci_unit_p)
- int [ng_hci_process_command_complete](#) (ng_hci_unit_p, struct mbuf *)
- int [ng_hci_process_command_status](#) (ng_hci_unit_p, struct mbuf *)
- void [ng_hci_process_command_timeout](#) (node_p, hook_p, void *, int)

7.32.1 Function Documentation

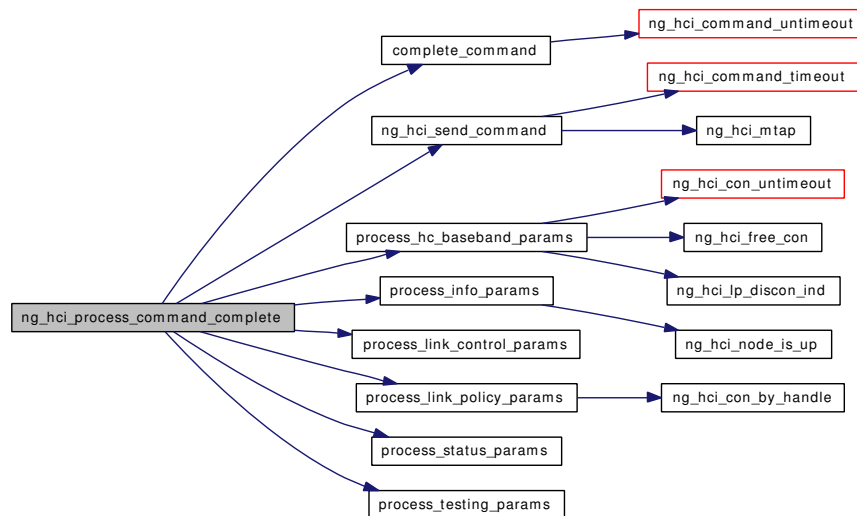
7.32.1.1 int [ng_hci_process_command_complete](#) (ng_hci_unit_p, struct mbuf *)

Definition at line 158 of file [ng_hci_cmds.c](#).

References [ng_hci_unit::buffer](#), [complete_command\(\)](#), [NG_FREE_M](#), [NG_HCI_BUFF_CMD_SET](#), [NG_HCI_ERR](#), [NG_HCI_M_PULLUP](#), [NG_HCI_OCF](#), [NG_HCI_OGF](#), [NG_HCI_OGF_BT_LOGO](#), [NG_HCI_OGF_HC_BASEBAND](#), [NG_HCI_OGF_INFO](#), [NG_HCI_OGF_LINK_CONTROL](#), [NG_HCI_OGF_LINK_POLICY](#), [NG_HCI_OGF_STATUS](#), [NG_HCI_OGF_TESTING](#), [NG_HCI_OGF_VENDOR](#), [ng_hci_send_command\(\)](#), [NG_NODE_NAME](#), [ng_hci_unit::node](#), [process_hc_baseband_params\(\)](#), [process_info_params\(\)](#), [process_link_control_params\(\)](#), [process_link_policy_params\(\)](#), [process_status_params\(\)](#), and [process_testing_params\(\)](#).

Referenced by [ng_hci_process_event\(\)](#).

Here is the call graph for this function:



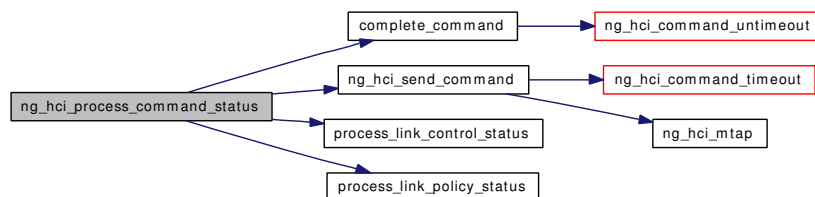
7.32.1.2 int ng_hci_process_command_status (ng_hci_unit_p, struct mbuf *)

Definition at line 260 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, complete_command(), NG_FREE_M, NG_HCI_BUFF_CMD_SET, NG_HCI_M_PULLUP, NG_HCI_OGF, NG_HCI_OGF_BT_LOGO, NG_HCI_OGF_HC_BASEBAND, NG_HCI_OGF_INFO, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OGF_LINK_POLICY, NG_HCI_OGF_STATUS, NG_HCI_OGF_TESTING, NG_HCI_OGF_VENDOR, ng_hci_send_command(), process_link_control_status(), and process_link_policy_status().

Referenced by ng_hci_process_event().

Here is the call graph for this function:



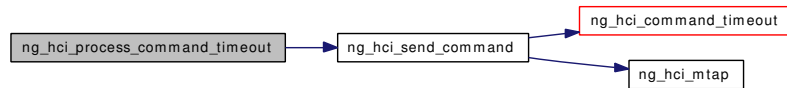
7.32.1.3 void ng_hci_process_command_timeout (node_p, hook_p, void *, int)

Definition at line 382 of file ng_hci_cmds.c.

References ng_hci_unit::buffer, ng_hci_unit::cmdq, NG_BT_MBUFQ_DEQUEUE, NG_FREE_M, NG_HCI_ALERT, NG_HCI_BUFF_CMD_SET, NG_HCI_ERR, NG_HCI_OCF, NG_HCI_OGF, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, ng_hci_unit::node, and ng_hci_unit::state.

Referenced by ng_hci_command_timeout().

Here is the call graph for this function:



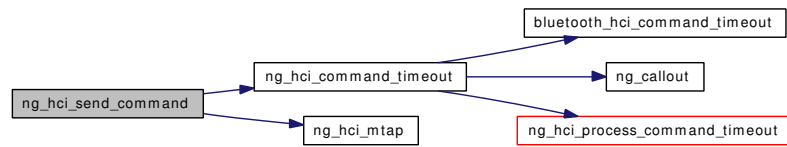
7.32.1.4 int ng_hci_send_command (ng_hci_unit_p)

Definition at line 85 of file `ng_hci_cmds.c`.

References `ng_hci_unit::buffer`, `ng_hci_unit::cmdq`, `ng_hci_unit::drv`, `NG_BT_MBUFQ_DRAIN`, `NG_BT_MBUFQ_FIRST`, `NG_HCI_BUFF_CMD_GET`, `NG_HCI_BUFF_CMD_USE`, `ng_hci_command_timeout()`, `NG_HCI_ERR`, `NG_HCI_HOOK_DRV`, `ng_hci_mtap()`, `NG_HCI_STAT_BYTES_SENT`, `NG_HCI_STAT_CMD_SENT`, `NG_HCI_UNIT_COMMAND_PENDING`, `NG_HCI_WARN`, `NG_HOOK_NOT_VALID`, `NG_NODE_NAME`, `NG_SEND_DATA_ONLY`, `ng_hci_unit::node`, `ng_hci_unit::stat`, and `ng_hci_unit::state`.

Referenced by `con_compl()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_process_command_complete()`, `ng_hci_process_command_status()`, `ng_hci_process_command_timeout()`, and `ng_hci_raw_rcvdata()`.

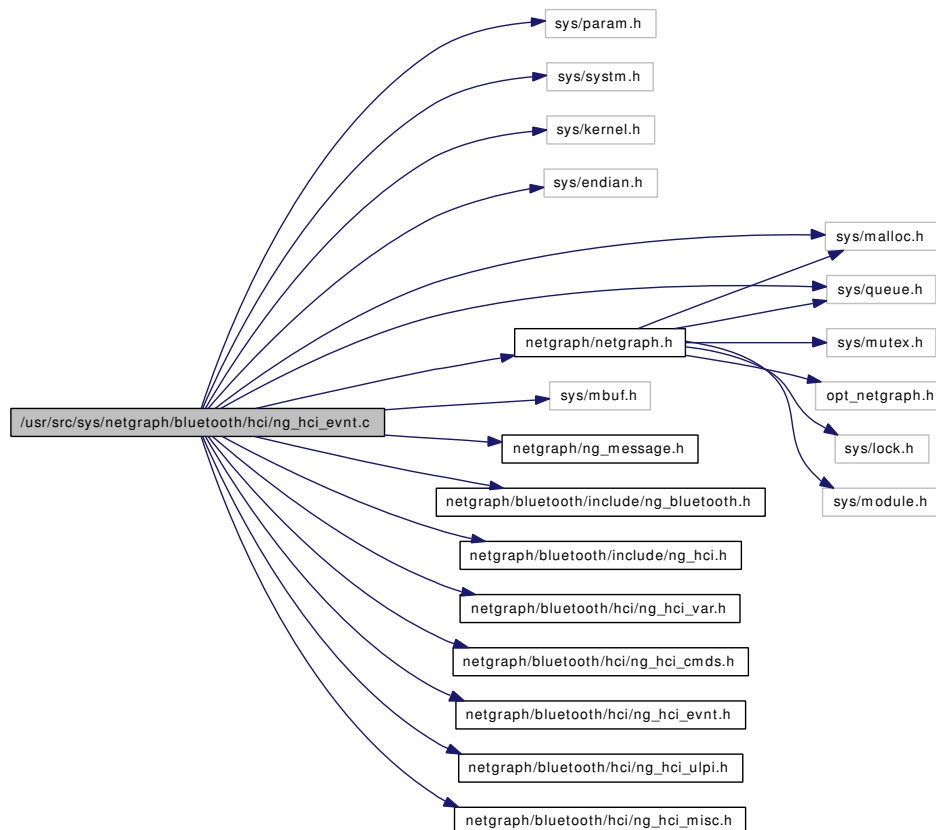
Here is the call graph for this function:



7.33 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/hci/ng_hci_var.h>
#include <netgraph/bluetooth/hci/ng_hci_cmds.h>
#include <netgraph/bluetooth/hci/ng_hci_evnt.h>
#include <netgraph/bluetooth/hci/ng_hci_ulpi.h>
#include <netgraph/bluetooth/hci/ng_hci_misc.h>
```

Include dependency graph for ng_hci_evnt.c:



Functions

- static int `inquiry_result` (`ng_hci_unit_p`, struct mbuf *)
- static int `con_compl` (`ng_hci_unit_p`, struct mbuf *)
- static int `con_req` (`ng_hci_unit_p`, struct mbuf *)
- static int `discon_compl` (`ng_hci_unit_p`, struct mbuf *)
- static int `encryption_change` (`ng_hci_unit_p`, struct mbuf *)
- static int `read_remote_features_compl` (`ng_hci_unit_p`, struct mbuf *)
- static int `qos_setup_compl` (`ng_hci_unit_p`, struct mbuf *)
- static int `hardware_error` (`ng_hci_unit_p`, struct mbuf *)
- static int `role_change` (`ng_hci_unit_p`, struct mbuf *)
- static int `num_compl_pkts` (`ng_hci_unit_p`, struct mbuf *)
- static int `mode_change` (`ng_hci_unit_p`, struct mbuf *)
- static int `data_buffer_overflow` (`ng_hci_unit_p`, struct mbuf *)
- static int `read_clock_offset_compl` (`ng_hci_unit_p`, struct mbuf *)
- static int `qos_violation` (`ng_hci_unit_p`, struct mbuf *)
- static int `page_scan_mode_change` (`ng_hci_unit_p`, struct mbuf *)
- static int `page_scan_rep_mode_change` (`ng_hci_unit_p`, struct mbuf *)
- static int `sync_con_queue` (`ng_hci_unit_p`, `ng_hci_unit_con_p`, int)
- static int `send_data_packets` (`ng_hci_unit_p`, int, int)
- int `ng_hci_process_event` (`ng_hci_unit_p` unit, struct mbuf *event)
- void `ng_hci_send_data` (`ng_hci_unit_p` unit)

7.33.1 Function Documentation

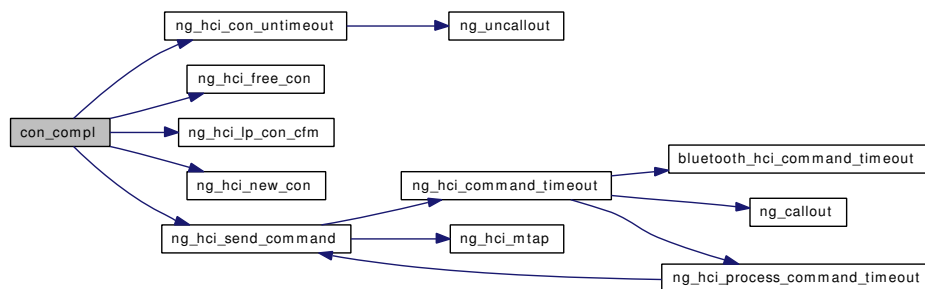
7.33.1.1 static int con_compl (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 429 of file ng_hci_evnt.c.

References ng_hci_unit_con::bdaddr, ng_hci_unit::cmdq, ng_hci_unit_con::con_handle, ng_hci_unit_con::encryption_mode, ng_hci_unit::features, ng_hci_unit::link_policy_mask, ng_hci_unit_con::link_type, NG_BT_MBUFQ_ENQUEUE, NG_FREE_M, NG_HCI_CMD_PKT, NG_HCI_CON_HANDLE, NG_HCI_CON_OPEN, ng_hci_con_untimeout(), NG_HCI_CON_W4_CONN_COMPLETE, ng_hci_free_con(), NG_HCI_LINK_ACL, NG_HCI_LMP_HOLD_MODE, NG_HCI_LMP_PARK_MODE, NG_HCI_LMP_SNIFF_MODE, NG_HCI_LMP_SWITCH, ng_hci_lp_con_cfm(), NG_HCI_M_PULLUP, ng_hci_new_con(), NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS, NG_HCI_OGF_LINK_POLICY, NG_HCI_OPCODE, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, packed, ng_hci_unit::role_switch, ng_hci_unit::state, and ng_hci_unit_con::state.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



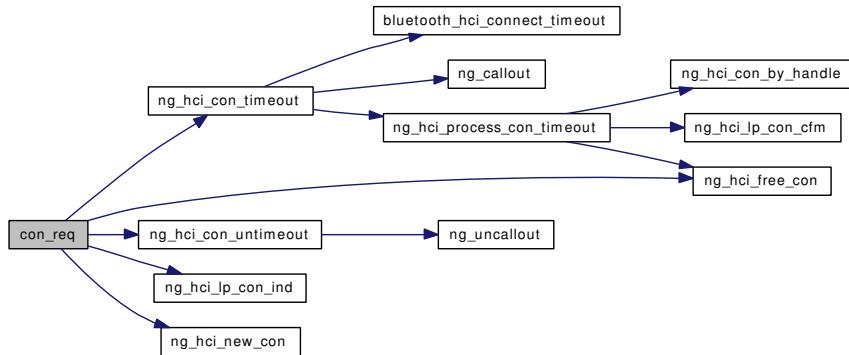
7.33.1.2 static int con_req (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 555 of file ng_hci_evnt.c.

References ng_hci_unit_con::bdaddr, ng_hci_unit_con::link_type, NG_FREE_M, ng_hci_con_timeout(), ng_hci_con_untimeout(), NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), ng_hci_lp_con_ind(), NG_HCI_M_PULLUP, ng_hci_new_con(), and ng_hci_unit_con::state.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



7.33.1.3 static int data_buffer_overflow (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 973 of file ng_hci_evnt.c.

References NG_FREE_M, NG_HCI_ALERT, NG_HCI_LINK_ACL, NG_NODE_NAME, and ng_hci_unit::node.

Referenced by ng_hci_process_event().

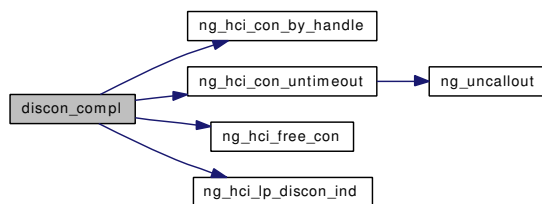
7.33.1.4 static int discon_compl (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 640 of file ng_hci_evnt.c.

References ng_hci_unit_con::flags, NG_FREE_M, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_CON_TIMEOUT_PENDING, ng_hci_con_untimeout(), ng_hci_free_con(), ng_hci_lp_discon_ind(), NG_HCI_M_PULLUP, NG_NODE_NAME, and ng_hci_unit::node.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



7.33.1.5 static int encryption_change (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 686 of file ng_hci_evnt.c.

References ng_hci_unit_con::encryption_mode, ng_hci_unit_con::link_type, NG_FREE_M, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_ENCRYPTION_MODE_NONE, NG_HCI_ENCRYPTION_MODE_P2P, NG_HCI_ERR, NG_HCI_LINK_ACL, NG_HCI_M_PULLUP, NG_NODE_NAME, and ng_hci_unit::node.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



7.33.1.6 static int hardware_error (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 824 of file ng_hci_evnt.c.

References NG_FREE_M, NG_HCI_ALERT, NG_NODE_NAME, and ng_hci_unit::node.

Referenced by ng_hci_process_event().

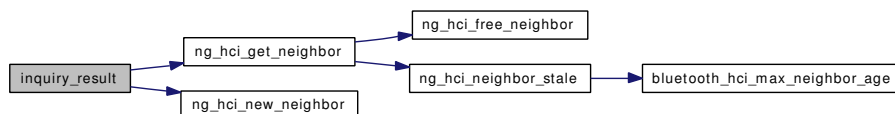
7.33.1.7 static int inquiry_result (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 369 of file ng_hci_evnt.c.

References ng_hci_neighbor::bdaddr, ng_hci_neighbor::clock_offset, NG_FREE_M, NG_HCI_CLASS_SIZE, ng_hci_get_neighbor(), NG_HCI_M_PULLUP, ng_hci_new_neighbor(), ng_hci_neighbor::page_scan_mode, ng_hci_neighbor::page_scan_rep_mode, and ng_hci_neighbor::updated.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



7.33.1.8 static int mode_change (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 932 of file ng_hci_evnt.c.

References ng_hci_unit_con::link_type, ng_hci_unit_con::mode, NG_FREE_M, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_ERR, NG_HCI_LINK_ACL, NG_HCI_M_PULLUP, NG_NODE_NAME, and ng_hci_unit::node.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



7.33.1.9 int ng_hci_process_event (ng_hci_unit_p unit, struct mbuf * event)

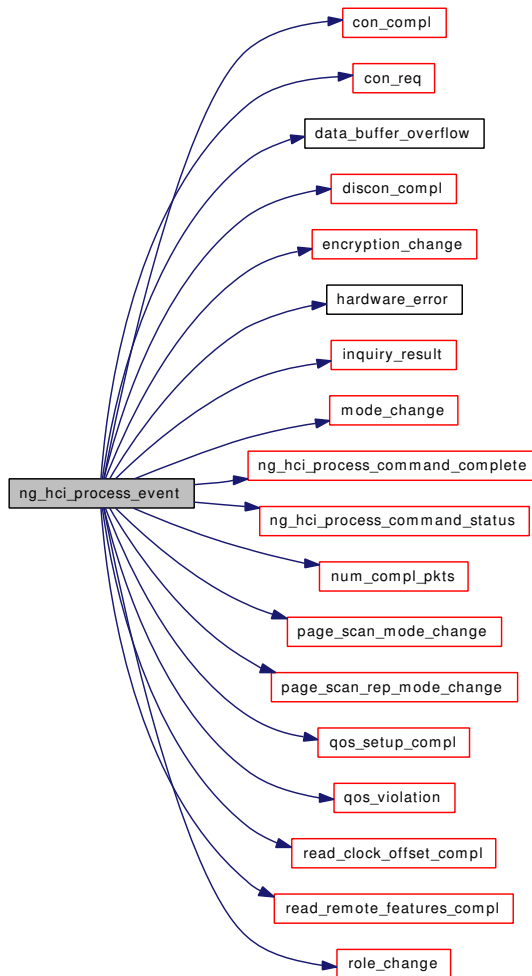
Definition at line 85 of file ng_hci_evnt.c.

References con_compl(), con_req(), data_buffer_overflow(), discon_compl(), encryption_change(), hardware_error(), inquiry_result(), mode_change(), NG_FREE_M, NG_HCI_EVENT_AUTH_COMPL, NG_HCI_EVENT_BT_LOGO, NG_HCI_EVENT_CHANGE_CON_LINK_KEY_COMPL,

NG_HCI_EVENT_COMMAND_COMPL, NG_HCI_EVENT_COMMAND_STATUS, NG_HCI_EVENT_CON_COMPL, NG_HCI_EVENT_CON_PKT_TYPE_CHANGED, NG_HCI_EVENT_CON_REQ, NG_HCI_EVENT_DATA_BUFFER_OVERFLOW, NG_HCI_EVENT_DISCON_COMPL, NG_HCI_EVENT_ENCRYPTION_CHANGE, NG_HCI_EVENT_FLUSH_OCCUR, NG_HCI_EVENT_HARDWARE_ERROR, NG_HCI_EVENT_INQUIRY_COMPL, NG_HCI_EVENT_INQUIRY_RESULT, NG_HCI_EVENT_LINK_KEY_NOTIFICATION, NG_HCI_EVENT_LINK_KEY_REQ, NG_HCI_EVENT_LOOPBACK_COMMAND, NG_HCI_EVENT_MASTER_LINK_KEY_COMPL, NG_HCI_EVENT_MAX_SLOT_CHANGE, NG_HCI_EVENT_MODE_CHANGE, NG_HCI_EVENT_NUM_COMPL_PKTS, NG_HCI_EVENT_PAGE_SCAN_MODE_CHANGE, NG_HCI_EVENT_PAGE_SCAN_REP_MODE_CHANGE, NG_HCI_EVENT_PIN_CODE_REQ, NG_HCI_EVENT_QOS_SETUP_COMPL, NG_HCI_EVENT_QOS_VIOLATION, NG_HCI_EVENT_READ_CLOCK_OFFSET_COMPL, NG_HCI_EVENT_READ_REMOTE_FEATURES_COMPL, NG_HCI_EVENT_READ_REMOTE_VER_INFO_COMPL, NG_HCI_EVENT_REMOTE_NAME_REQ_COMPL, NG_HCI_EVENT_RETURN_LINK_KEYS, NG_HCI_EVENT_ROLE_CHANGE, NG_HCI_EVENT_VENDOR, NG_HCI_INFO, NG_HCI_M_PULLUP, ng_hci_process_command_complete(), ng_hci_process_command_status(), NG_NODE_NAME, ng_hci_unit::node, num_compl_pkts(), page_scan_mode_change(), page_scan_rep_mode_change(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), read_remote_features_compl(), and role_change().

Referenced by ng_hci_drv_rcvdata().

Here is the call graph for this function:



7.33.1.10 void ng_hci_send_data (ng_hci_unit_p unit)

Definition at line 211 of file ng_hci_evnt.c.

References ng_hci_unit::buffer, NG_HCI_BUFF_ACL_AVAIL, NG_HCI_BUFF_ACL_USE, NG_HCI_BUFF_SCO_AVAIL, NG_HCI_BUFF_SCO_USE, NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HCI_LINK_SCO, NG_HCI_STAT_ACL_SENT, NG_HCI_STAT_SCO_SENT, NG_NODE_NAME, ng_hci_unit::node, send_data_packets(), and ng_hci_unit::stat.

Referenced by ng_hci_acl_rcvdata(), ng_hci_sco_rcvdata(), and num_compl_pkts().

Here is the call graph for this function:

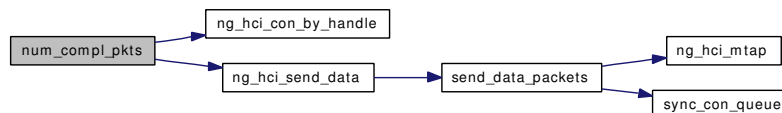
**7.33.1.11 static int num_compl_pkts (ng_hci_unit_p, struct mbuf *) [static]**

Definition at line 874 of file ng_hci_evnt.c.

References ng_hci_unit::buffer, ng_hci_unit_con::con_handle, ng_hci_unit_con::link_type, NG_FREE_M, NG_HCI_ALERT, NG_HCI_BUFF_ACL_FREE, NG_HCI_BUFF_SCO_FREE, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_LINK_ACL, NG_HCI_M_PULLUP, ng_hci_send_data(), NG_HCI_WARN, NG_NODE_NAME, ng_hci_unit::node, and ng_hci_unit_con::pending.

Referenced by ng_hci_process_event().

Here is the call graph for this function:

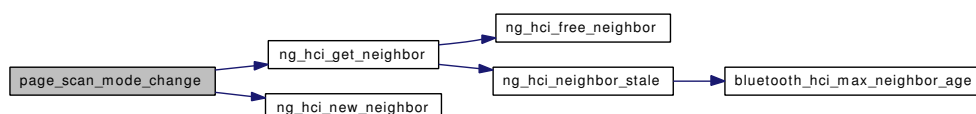
**7.33.1.12 static int page_scan_mode_change (ng_hci_unit_p, struct mbuf *) [static]**

Definition at line 1079 of file ng_hci_evnt.c.

References ng_hci_neighbor::bdaddr, ng_hci_unit::bdaddr, NG_FREE_M, ng_hci_get_neighbor(), NG_HCI_M_PULLUP, ng_hci_new_neighbor(), ng_hci_neighbor::page_scan_mode, and ng_hci_neighbor::updated.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



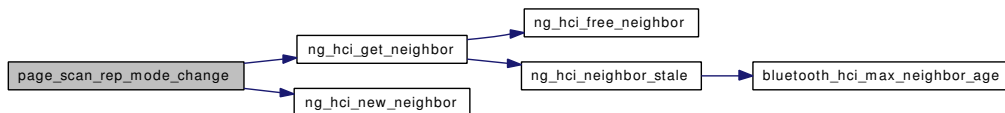
7.33.1.13 static int page_scan_rep_mode_change (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 1113 of file ng_hci_evnt.c.

References ng_hci_neighbor::bdaddr, ng_hci_unit::bdaddr, NG_FREE_M, ng_hci_get_neighbor(), NG_HCI_M_PULLUP, ng_hci_new_neighbor(), ng_hci_neighbor::page_scan_rep_mode, and ng_hci_neighbor::updated.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



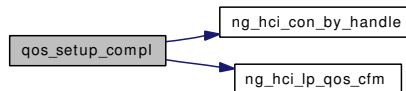
7.33.1.14 static int qos_setup_compl (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 782 of file ng_hci_evnt.c.

References ng_hci_unit_con::link_type, NG_FREE_M, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_CON_OPEN, NG_HCI_LINK_ACL, ng_hci_lp_qos_cfm(), NG_HCI_M_PULLUP, NG_NODE_NAME, ng_hci_unit::node, and ng_hci_unit_con::state.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



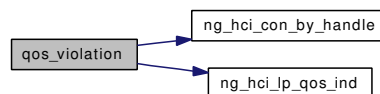
7.33.1.15 static int qos_violation (ng_hci_unit_p, struct mbuf *) [static]

Definition at line 1038 of file ng_hci_evnt.c.

References ng_hci_unit_con::link_type, NG_FREE_M, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_CON_OPEN, NG_HCI_LINK_ACL, ng_hci_lp_qos_ind(), NG_HCI_M_PULLUP, NG_NODE_NAME, ng_hci_unit::node, and ng_hci_unit_con::state.

Referenced by ng_hci_process_event().

Here is the call graph for this function:



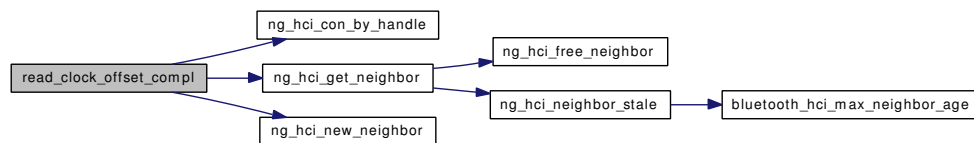
7.33.1.16 `static int read_clock_offset_compl (ng_hci_unit_p, struct mbuf *)` [static]

Definition at line 987 of file ng_hci_evnt.c.

References `ng_hci_neighbor::bdaddr`, `ng_hci_unit_con::bdaddr`, `ng_hci_neighbor::clock_offset`, `NG_FREE_M`, `NG_HCI_ALERT`, `ng_hci_con_by_handle()`, `NG_HCI_CON_HANDLE`, `NG_HCI_ERR`, `ng_hci_get_neighbor()`, `NG_HCI_M_PULLUP`, `ng_hci_new_neighbor()`, `NG_NODE_NAME`, `ng_hci_unit::node`, and `ng_hci_neighbor::updated`.

Referenced by `ng_hci_process_event()`.

Here is the call graph for this function:

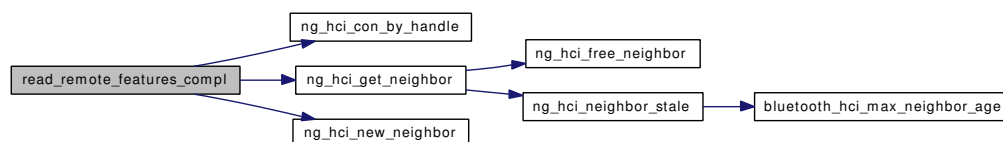
**7.33.1.17** `static int read_remote_features_compl (ng_hci_unit_p, struct mbuf *)` [static]

Definition at line 730 of file ng_hci_evnt.c.

References `ng_hci_neighbor::bdaddr`, `ng_hci_unit_con::bdaddr`, `ng_hci_neighbor::features`, `NG_FREE_M`, `NG_HCI_ALERT`, `ng_hci_con_by_handle()`, `NG_HCI_CON_HANDLE`, `NG_HCI_ERR`, `ng_hci_get_neighbor()`, `NG_HCI_M_PULLUP`, `ng_hci_new_neighbor()`, `NG_NODE_NAME`, `ng_hci_unit::node`, and `ng_hci_neighbor::updated`.

Referenced by `ng_hci_process_event()`.

Here is the call graph for this function:

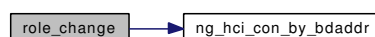
**7.33.1.18** `static int role_change (ng_hci_unit_p, struct mbuf *)` [static]

Definition at line 837 of file ng_hci_evnt.c.

References `ng_hci_unit::bdaddr`, `NG_FREE_M`, `NG_HCI_ALERT`, `ng_hci_con_by_bdaddr()`, `NG_HCI_ERR`, `NG_HCI_LINK_ACL`, `NG_HCI_M_PULLUP`, `NG_NODE_NAME`, `ng_hci_unit::node`, and `ng_hci_unit_con::role`.

Referenced by `ng_hci_process_event()`.

Here is the call graph for this function:



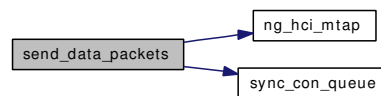
7.33.1.19 `static int send_data_packets (ng_hci_unit_p, int, int)` [static]

Definition at line 247 of file `ng_hci_evnt.c`.

References `ng_hci_unit_con::conq`, `ng_hci_unit::drv`, `ng_hci_unit_con::link_type`, `NG_BT_ITEMQ_DEQUEUE`, `NG_BT_ITEMQ_LEN`, `NG_FREE_ITEM`, `NG_FWD_ITEM_HOOK`, `NG_HCI_ERR`, `NG_HCI_HOOK_DRV`, `NG_HCI_INFO`, `ng_hci_mtap()`, `NG_HCI_STAT_BYTES_SENT`, `NG_HCI_UNIT_READY`, `NG_HOOK_IS_VALID`, `NG_NODE_NAME`, `NGI_M`, `ng_hci_unit::node`, `ng_hci_unit_con::pending`, `ng_hci_unit::stat`, `ng_hci_unit::state`, and `sync_con_queue()`.

Referenced by `ng_hci_send_data()`.

Here is the call graph for this function:

**7.33.1.20** `static int sync_con_queue (ng_hci_unit_p, ng_hci_unit_con_p, int)` [static]

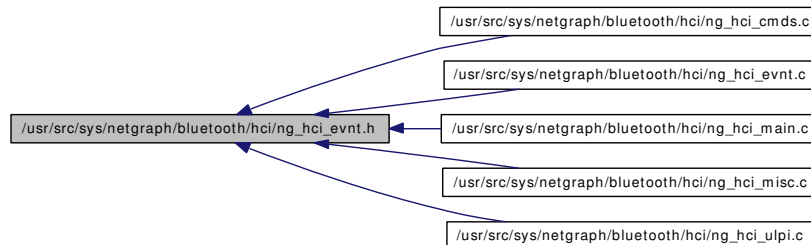
Definition at line 342 of file `ng_hci_evnt.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::con_handle`, `ng_mesg::data`, `ng_hci_unit_con::link_type`, `NG_HCI_LINK_ACL`, `NG_HOOK_NOT_VALID`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_SYNC_CON_QUEUE`, `ng_hci_unit::node`, and `ng_hci_unit::sco`.

Referenced by `send_data_packets()`.

7.34 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int ng_hci_process_event (ng_hci_unit_p, struct mbuf *)`
- `void ng_hci_send_data (ng_hci_unit_p)`

7.34.1 Function Documentation

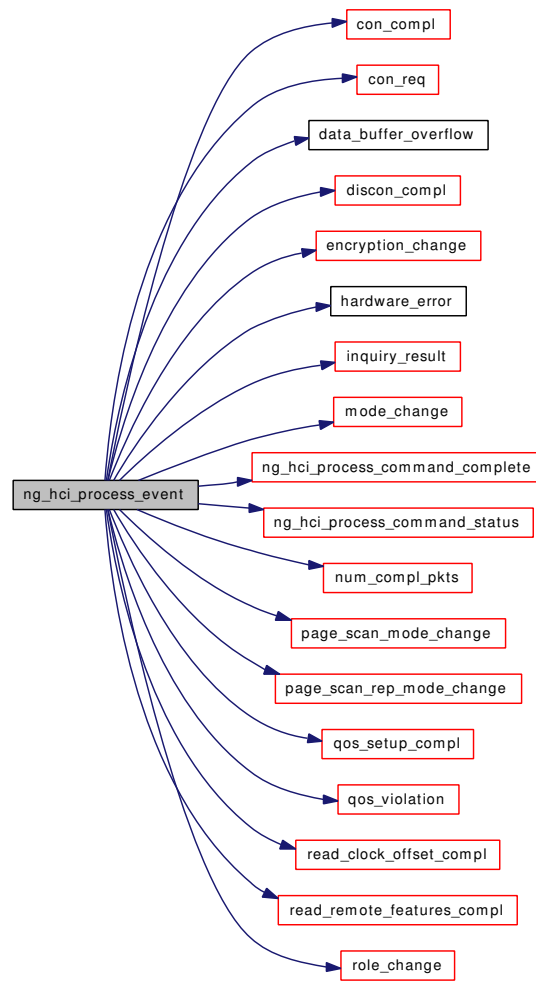
7.34.1.1 `int ng_hci_process_event (ng_hci_unit_p, struct mbuf *)`

Definition at line 85 of file `ng_hci_evnt.c`.

References `con_compl()`, `con_req()`, `data_buffer_overflow()`, `discon_compl()`, `encryption_change()`, `hardware_error()`, `inquiry_result()`, `mode_change()`, `NG_FREE_M`, `NG_HCI_EVENT_AUTH_COMPL`, `NG_HCI_EVENT_BT_LOGO`, `NG_HCI_EVENT_CHANGE_CON_LINK_KEY_COMPL`, `NG_HCI_EVENT_COMMAND_COMPL`, `NG_HCI_EVENT_COMMAND_STATUS`, `NG_HCI_EVENT_CON_COMPL`, `NG_HCI_EVENT_CON_PKT_TYPE_CHANGED`, `NG_HCI_EVENT_CON_REQ`, `NG_HCI_EVENT_DATA_BUFFER_OVERFLOW`, `NG_HCI_EVENT_DISCON_COMPL`, `NG_HCI_EVENT_ENCRYPTION_CHANGE`, `NG_HCI_EVENT_FLUSH_OCCUR`, `NG_HCI_EVENT_HARDWARE_ERROR`, `NG_HCI_EVENT_INQUIRY_COMPL`, `NG_HCI_EVENT_INQUIRY_RESULT`, `NG_HCI_EVENT_LINK_KEY_NOTIFICATION`, `NG_HCI_EVENT_LINK_KEY_REQ`, `NG_HCI_EVENT_LOOPBACK_COMMAND`, `NG_HCI_EVENT_MASTER_LINK_KEY_COMPL`, `NG_HCI_EVENT_MAX_SLOT_CHANGE`, `NG_HCI_EVENT_MODE_CHANGE`, `NG_HCI_EVENT_NUM_COMPL_PKTS`, `NG_HCI_EVENT_PAGE_SCAN_MODE_CHANGE`, `NG_HCI_EVENT_PAGE_SCAN_REP_MODE_CHANGE`, `NG_HCI_EVENT_PIN_CODE_REQ`, `NG_HCI_EVENT_QOS_SETUP_COMPL`, `NG_HCI_EVENT_QOS_VIOLATION`, `NG_HCI_EVENT_READ_CLOCK_OFFSET_COMPL`, `NG_HCI_EVENT_READ_REMOTE_FEATURES_COMPL`, `NG_HCI_EVENT_READ_REMOTE_VER_INFO_COMPL`, `NG_HCI_EVENT_REMOTE_NAME_REQ_COMPL`, `NG_HCI_EVENT_RETURN_LINK_KEYS`, `NG_HCI_EVENT_ROLE_CHANGE`, `NG_HCI_EVENT_VENDOR`, `NG_HCI_INFO`, `NG_HCI_M_PULLUP`, `ng_hci_process_command_complete()`, `ng_hci_process_command_status()`, `NG_NODE_NAME`, `ng_hci_unit::node`, `num_compl_pkts()`, `page_scan_mode_change()`, `page_scan_rep_mode_change()`, `qos_setup_compl()`, `qos_violation()`, `read_clock_offset_compl()`, `read_remote_features_compl()`, and `role_change()`.

Referenced by `ng_hci_drv_rcvdata()`.

Here is the call graph for this function:



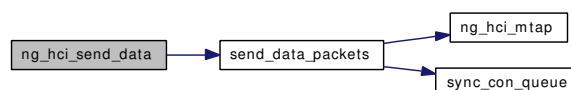
7.34.1.2 void ng_hci_send_data (ng_hci_unit_p)

Definition at line 211 of file ng_hci_evnt.c.

References ng_hci_unit::buffer, NG_HCI_BUFF_ACL_AVAIL, NG_HCI_BUFF_ACL_USE, NG_HCI_BUFF_SCO_AVAIL, NG_HCI_BUFF_SCO_USE, NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HCI_LINK_SCO, NG_HCI_STAT_ACL_SENT, NG_HCI_STAT_SCO_SENT, NG_NODE_NAME, ng_hci_unit::node, send_data_packets(), and ng_hci_unit::stat.

Referenced by ng_hci_acl_rcvdata(), ng_hci_sco_rcvdata(), and num_compl_pkts().

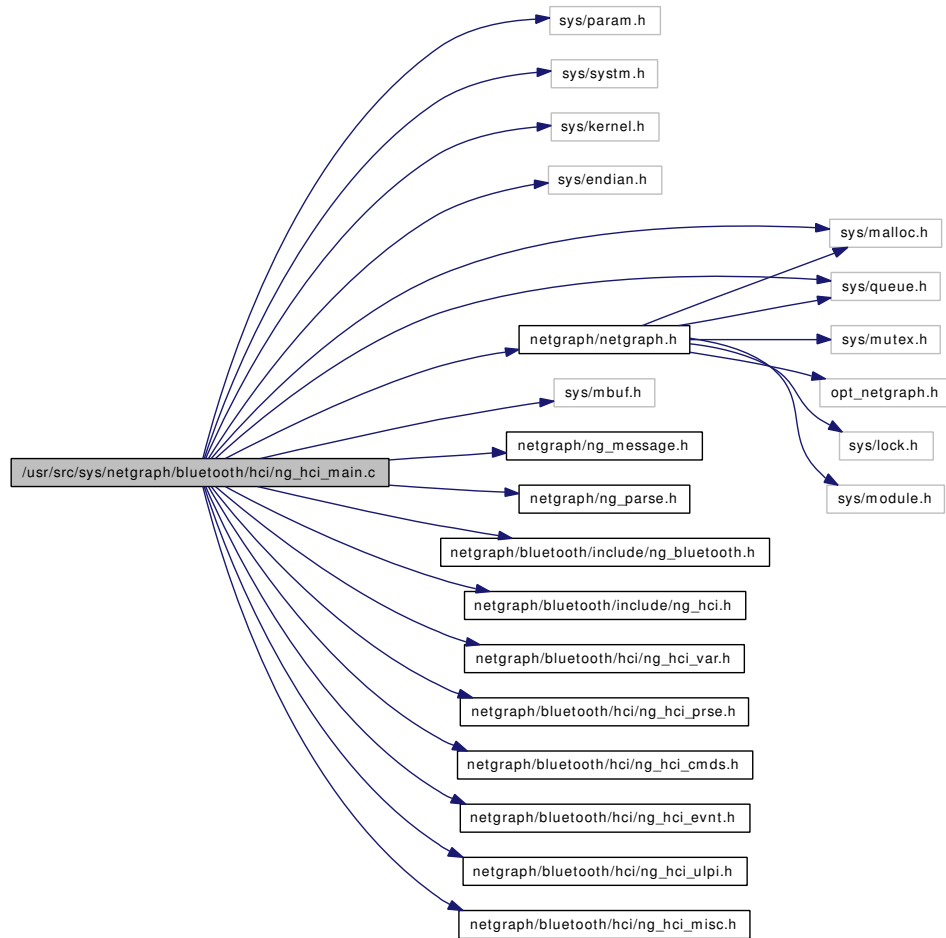
Here is the call graph for this function:



7.35 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_main.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/hci/ng_hci_var.h>
#include <netgraph/bluetooth/hci/ng_hci_prse.h>
#include <netgraph/bluetooth/hci/ng_hci_cmds.h>
#include <netgraph/bluetooth/hci/ng_hci_evnt.h>
#include <netgraph/bluetooth/hci/ng_hci_ulpi.h>
#include <netgraph/bluetooth/hci/ng_hci_misc.h>
```

Include dependency graph for ng_hci_main.c:



Defines

- #define [M_NETGRAPH_HCI](#) M_NETGRAPH

Functions

- [NETGRAPH_INIT](#) (hci,&typestruct)
- [MODULE_VERSION](#) (ng_hci, NG_BLUETOOTH_VERSION)
- [MODULE_DEPEND](#) (ng_hci, ng_bluetooth, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION)
- static int [ng_hci_constructor](#) (node_p node)
- static int [ng_hci_shutdown](#) (node_p node)
- static int [ng_hci_newhook](#) (node_p node, hook_p hook, char const *name)
- static int [ng_hci_connect](#) (hook_p hook)
- static int [ng_hci_disconnect](#) (hook_p hook)
- static int [ng_hci_default_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_hci_upper_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_hci_drv_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_hci_acl_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_hci_sco_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_hci_raw_rcvdata](#) (hook_p hook, item_p item)

Variables

- static `ng_constructor_t` `ng_hci_constructor`
- static `ng_shutdown_t` `ng_hci_shutdown`
- static `ng_newhook_t` `ng_hci_newhook`
- static `ng_connect_t` `ng_hci_connect`
- static `ng_disconnect_t` `ng_hci_disconnect`
- static `ng_rcvmsg_t` `ng_hci_default_rcvmsg`
- static `ng_rcvmsg_t` `ng_hci_upper_rcvmsg`
- static `ng_rcvdata_t` `ng_hci_drv_rcvdata`
- static `ng_rcvdata_t` `ng_hci_acl_rcvdata`
- static `ng_rcvdata_t` `ng_hci_sco_rcvdata`
- static `ng_rcvdata_t` `ng_hci_raw_rcvdata`
- static struct `ng_type` `typestruct`

7.35.1 Define Documentation

7.35.1.1 #define M_NETGRAPH_HCI M_NETGRAPH

Definition at line 63 of file `ng_hci_main.c`.

Referenced by `ng_hci_constructor()`, `ng_hci_free_con()`, `ng_hci_free_neighbor()`, `ng_hci_new_con()`, `ng_hci_new_neighbor()`, and `ng_hci_shutdown()`.

7.35.2 Function Documentation

7.35.2.1 MODULE_DEPEND (ng_hci, ng_bluetooth, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION)

7.35.2.2 MODULE_VERSION (ng_hci, NG_BLUETOOTH_VERSION)

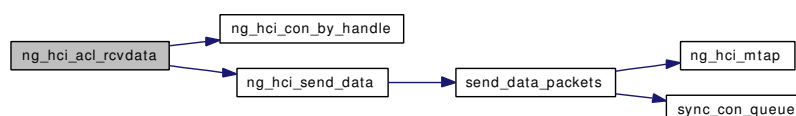
7.35.2.3 NETGRAPH_INIT (hci, & typestruct)

7.35.2.4 static int ng_hci_acl_rcvdata (hook_p hook, item_p item) [static]

Definition at line 772 of file `ng_hci_main.c`.

References `ng_hci_unit::buffer`, `ng_hci_unit_con::conq`, `ng_hci_unit_con::link_type`, `NG_BT_ITEMQ_DROP`, `NG_BT_ITEMQ_ENQUEUE`, `NG_BT_ITEMQ_FULL`, `NG_BT_ITEMQ_LEN`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HCI_ACL_DATA_PKT`, `NG_HCI_ALERT`, `NG_HCI_BUFF_ACL_SIZE`, `ng_hci_con_by_handle()`, `NG_HCI_CON_HANDLE`, `NG_HCI_CON_OPEN`, `NG_HCI_ERR`, `NG_HCI_LINK_ACL`, `NG_HCI_M_PULLUP`, `ng_hci_send_data()`, `NG_HOOK_NODE`, `NG_NODE_NAME`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `NGI_M`, `ng_hci_unit::node`, and `ng_hci_unit_con::state`.

Here is the call graph for this function:



7.35.2.5 static int ng_hci_connect (**hook_p** hook) [static]

Definition at line 214 of file ng_hci_main.c.

References ng_hci_unit::acl, ng_hci_unit::drv, ng_hci_acl_rcvdata, ng_hci_node_is_up(), ng_hci_raw_rcvdata, ng_hci_sco_rcvdata, NG_HCI_UNIT_CONNECTED, ng_hci_upper_rcvmsg, NG_HOOK_NODE, NG_HOOK_SET_RCVDATA, NG_HOOK_SET_RCVMSG, NG_NODE_PRIVATE, ng_send_fn, ng_hci_unit::node, ng_hci_unit::raw, ng_hci_unit::sco, and ng_hci_unit::state.

Here is the call graph for this function:



7.35.2.6 static int ng_hci_constructor (**node_p** node) [static]

Definition at line 108 of file ng_hci_main.c.

References M_NETGRAPH_HCI, NG_BT_MBUFQ_INIT, ng_callout_init, NG_HCI_BUFF_ACL_SET, NG_HCI_BUFF_CMD_SET, NG_HCI_BUFF_SCO_SET, NG_HCI_CMD_QUEUE_LEN, NG_HCI_WARN_LEVEL, NG_NODE_FORCE_WRITER, and NG_NODE_SET_PRIVATE.

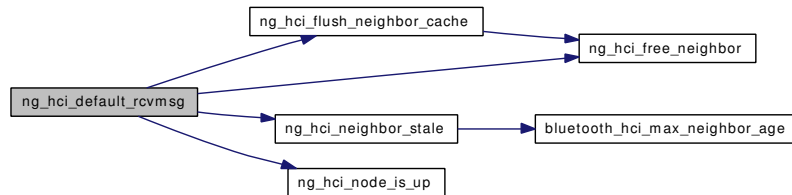
7.35.2.7 static int ng_hci_default_rcvmsg (**node_p** node, **item_p** item, **hook_p** lasthook) [static]

Definition at line 278 of file ng_hci_main.c.

References ng_hci_unit::acl, ng_hci_node_buffer_ep::acl_free, ng_hci_node_buffer_ep::acl_pkts, ng_hci_node_buffer_ep::acl_size, ng_msg::ng_msghdr::arglen, ng_hci_unit_con::bdaddr, ng_hci_neighbor::bdaddr, ng_hci_unit::bdaddr, ng_hci_unit::buffer, ng_hci_neighbor::clock_offset, ng_msg::ng_msghdr::cmd, ng_hci_node_buffer_ep::cmd_free, ng_hci_unit::cmdq, ng_hci_unit_con::con_handle, ng_hci_unit_con::conq, ng_msg::data, ng_hci_unit::debug, ng_hci_unit::drv, ng_hci_unit_con::encryption_mode, ng_hci_neighbor::features, ng_hci_unit::features, ng_msg::header, ng_hci_unit::link_policy_mask, ng_hci_unit_con::link_type, ng_hci_unit_con::mode, NG_BT_ITEMQ_LEN, NG_BT_MBUFQ_LEN, NG_FREE_MSG, NG_HCI_BDADDR_ANY, NG_HCI_BUFF_ACL_AVAIL, NG_HCI_BUFF_ACL_SIZE, NG_HCI_BUFF_ACL_TOTAL, NG_HCI_BUFF_CMD_GET, NG_HCI_BUFF_SCO_AVAIL, NG_HCI_BUFF_SCO_SIZE, NG_HCI_BUFF_SCO_TOTAL, ng_hci_flush_neighbor_cache(), ng_hci_free_neighbor(), NG_HCI_HOOK_ACL, NG_HCI_HOOK_DRV, NG_HCI_HOOK_RAW, NG_HCI_HOOK_SCO, NG_HCI_MAX_CON_NUM, NG_HCI_MAX_NEIGHBOR_NUM, ng_hci_neighbor_stale(), ng_hci_node_is_up(), NG_HCI_STAT_RESET, NG_HCI_UNIT_INITED, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NG_TEXTRESPONSE, NGI_GET_MSG, NGM_GENERIC_COOKIE, NGM_HCI_COOKIE, NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE, NGM_HCI_NODE_GET_BDADDR, NGM_HCI_NODE_GET_BUFFER, NGM_HCI_NODE_GET_CON_LIST, NGM_HCI_NODE_GET_DEBUG, NGM_HCI_NODE_GET_FEATURES, NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK, NGM_HCI_NODE_GET_NEIGHBOR_CACHE, NGM_HCI_NODE_GET_PACKET_MASK, NGM_HCI_NODE_GET_ROLE_SWITCH, NGM_HCI_NODE_GET_STAT, NGM_HCI_NODE_GET_STATE, NGM_HCI_NODE_INIT, NGM_HCI_NODE_RESET_STAT, NGM_HCI_NODE_SET_DEBUG, NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK, NGM_HCI_NODE_SET_PACKET_MASK, NGM_HCI_NODE_SET_ROLE_SWITCH, NGM_TEXT_STATUS, ng_hci_unit::node, ng_hci_unit::packet_mask, ng_hci_neighbor::page_scan_mode, ng_hci_neighbor::page_scan_rep_mode, ng_hci_unit_con::pending, ng_hci_unit::raw, ng_hci_unit_con::role, ng_hci_unit::role_switch, ng_hci_unit::sco,

ng_hci_node_buffer_ep::sco_free, ng_hci_node_buffer_ep::sco_pkts, ng_hci_node_buffer_ep::sco_size, ng_hci_unit::stat, ng_hci_unit_con::state, ng_hci_unit::state, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:

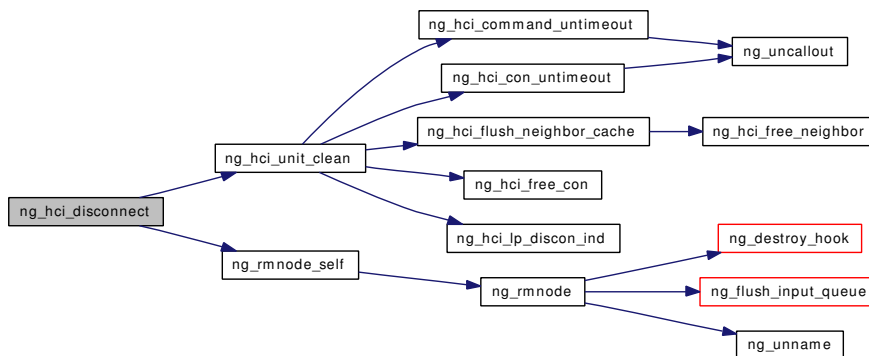


7.35.2.8 static int ng_hci_disconnect (hook_p hook) [static]

Definition at line 242 of file ng_hci_main.c.

References ng_hci_unit::acl, ng_hci_unit::drv, ng_hci_unit_clean(), NG_HCI_UNIT_CONNECTED, NG_HCI_UNIT_INITED, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), ng_hci_unit::raw, ng_hci_unit::sco, and ng_hci_unit::state.

Here is the call graph for this function:

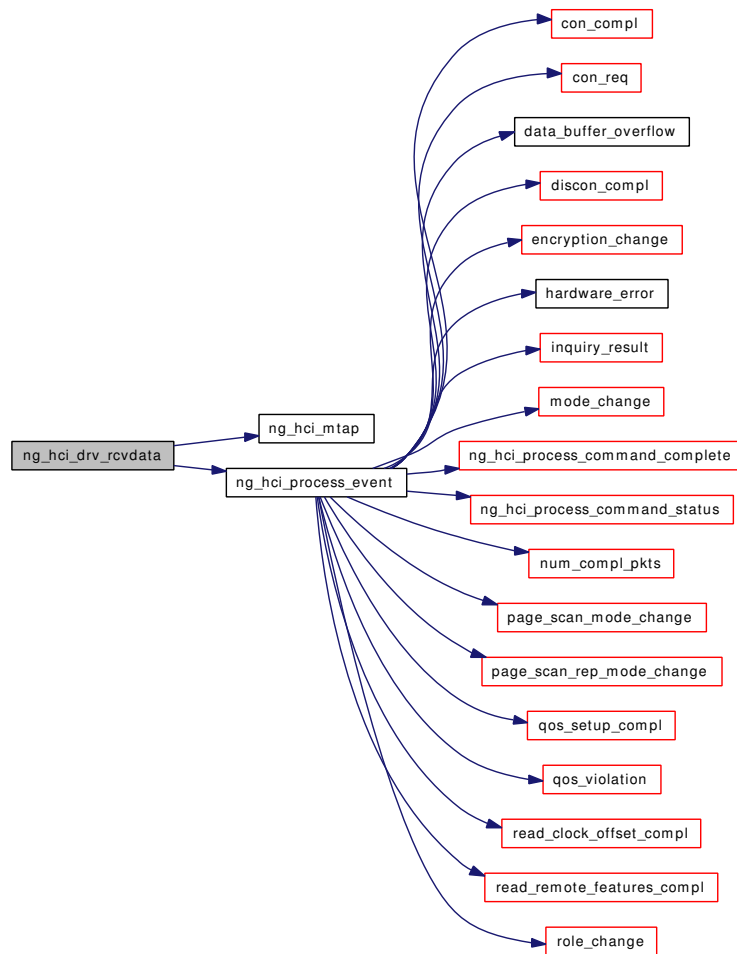


7.35.2.9 static int ng_hci_drv_rcvdata (hook_p hook, item_p item) [static]

Definition at line 689 of file ng_hci_main.c.

References ng_hci_unit::acl, NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_HCI_ACL_DATA_PKT, NG_HCI_ALERT, NG_HCI_EVENT_PKT, ng_hci_mtap(), ng_hci_process_event(), NG_HCI_SCO_DATA_PKT, NG_HCI_STAT_ACL_RECV, NG_HCI_STAT_BYTES_RECV, NG_HCI_STAT_EVNT_RECV, NG_HCI_STAT_SCO_RECV, NG_HCI_UNIT_READY, NG_HCI_WARN, NG_HOOK_NODE, NG_HOOK_NOT_VALID, NG_NODE_NAME, NG_NODE_PRIVATE, NGI_GET_M, NGI_M, ng_hci_unit::node, ng_hci_unit::sco, ng_hci_unit::stat, and ng_hci_unit::state.

Here is the call graph for this function:



7.35.2.10 `static int ng_hci_newhook (node_p node, hook_p hook, char const * name)` [static]

Definition at line 185 of file `ng_hci_main.c`.

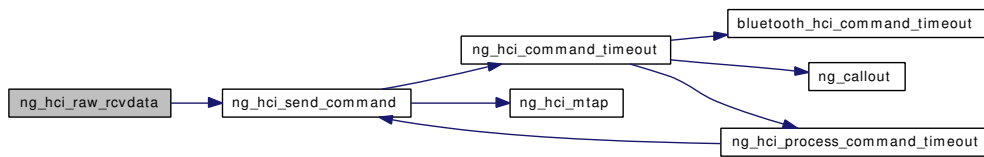
References `ng_hci_unit::acl`, `ng_hci_unit::drv`, `NG_HCI_HOOK_ACL`, `NG_HCI_HOOK_DRV`, `NG_HCI_HOOK_RAW`, `NG_HCI_HOOK_SCO`, `NG_NODE_PRIVATE`, `ng_hci_unit::raw`, and `ng_hci_unit::sco`.

7.35.2.11 `static int ng_hci_raw_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 1010 of file `ng_hci_main.c`.

References `ng_hci_unit::cmdq`, `NG_BT_MBUFQ_DROP`, `NG_BT_MBUFQ_ENQUEUE`, `NG_BT_MBUFQ_FULL`, `NG_BT_MBUFQ_LEN`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HCI_ALERT`, `NG_HCI_CMD_PKT`, `NG_HCI_M_PULLUP`, `ng_hci_send_command()`, `NG_HCI_UNIT_COMMAND_PENDING`, `NG_HOOK_NODE`, `NG_NODE_NAME`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `ng_hci_unit::node`, and `ng_hci_unit::state`.

Here is the call graph for this function:

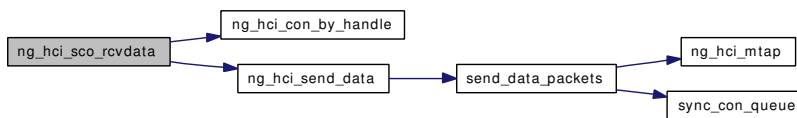


7.35.2.12 static int ng_hci_sco_rcvdata (hook_p hook, item_p item) [static]

Definition at line 891 of file ng_hci_main.c.

References ng_hci_unit::buffer, ng_hci_unit_con::conq, ng_hci_unit_con::link_type, NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_ITEMQ_LEN, NG_FREE_ITEM, NG_FREE_M, NG_HCI_ALERT, NG_HCI_BUFF_SCO_SIZE, ng_hci_con_by_handle(), NG_HCI_CON_HANDLE, NG_HCI_CON_OPEN, NG_HCI_ERR, NG_HCI_LINK_SCO, NG_HCI_M_PULLUP, NG_HCI_SCO_DATA_PKT, ng_hci_send_data(), NG_HOOK_NODE, NG_NODE_NAME, NG_NODE_PRIVATE, NGI_GET_M, NGI_M, ng_hci_unit::node, and ng_hci_unit_con::state.

Here is the call graph for this function:

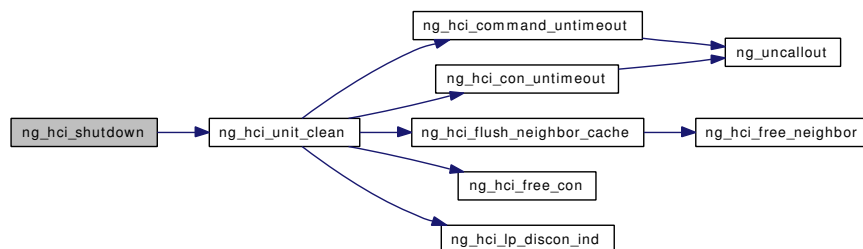


7.35.2.13 static int ng_hci_shutdown (node_p node) [static]

Definition at line 160 of file ng_hci_main.c.

References ng_hci_unit::cmdq, M_NETGRAPH_HCI, NG_BT_MBUFQ_DESTROY, ng_hci_unit_clean(), NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_hci_unit::node.

Here is the call graph for this function:

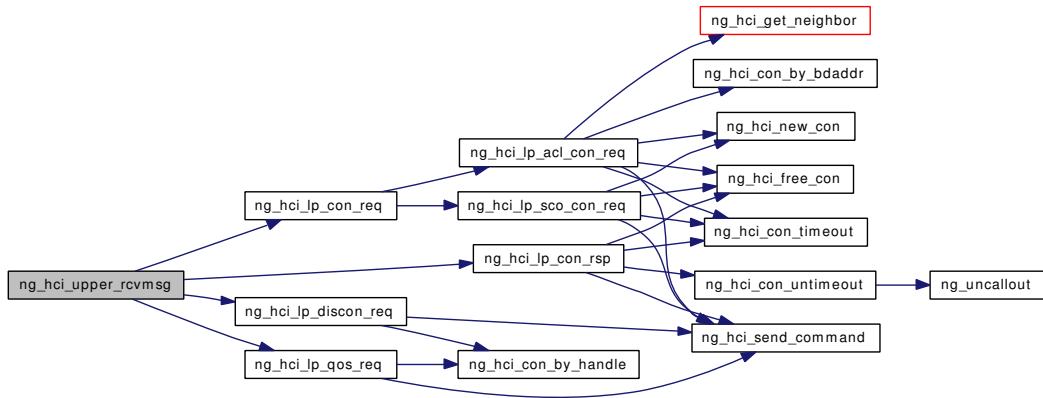


7.35.2.14 static int ng_hci_upper_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 645 of file ng_hci_main.c.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::header`, `ng_hci_default_rcvmsg`, `ng_hci_lp_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `NG_NODE_PRIVATE`, `NGI_MSG`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_CON_REQ`, `NGM_HCI_LP_CON_RSP`, `NGM_HCI_LP_DISCON_REQ`, `NGM_HCI_LP_QOS_REQ`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.35.3 Variable Documentation

7.35.3.1 `ng_rcvdata_t ng_hci_acl_rcvdata` [static]

Definition at line 75 of file `ng_hci_main.c`.

Referenced by `ng_hci_connect()`.

7.35.3.2 `ng_connect_t ng_hci_connect` [static]

Definition at line 70 of file `ng_hci_main.c`.

7.35.3.3 `ng_constructor_t ng_hci_constructor` [static]

Definition at line 67 of file `ng_hci_main.c`.

7.35.3.4 `ng_rcvmsg_t ng_hci_default_rcvmsg` [static]

Definition at line 72 of file `ng_hci_main.c`.

Referenced by `ng_hci_upper_rcvmsg()`.

7.35.3.5 `ng_disconnect_t ng_hci_disconnect` [static]

Definition at line 71 of file `ng_hci_main.c`.

7.35.3.6 `ng_rcvdata_t ng_hci_drv_rcvdata` [static]

Definition at line 74 of file `ng_hci_main.c`.

7.35.3.7 `ng_newhook_t ng_hci_newhook` [static]

Definition at line 69 of file `ng_hci_main.c`.

7.35.3.8 `ng_rcvdata_t ng_hci_raw_rcvdata` [static]

Definition at line 77 of file `ng_hci_main.c`.

Referenced by `ng_hci_connect()`.

7.35.3.9 `ng_rcvdata_t ng_hci_sco_rcvdata` [static]

Definition at line 76 of file `ng_hci_main.c`.

Referenced by `ng_hci_connect()`.

7.35.3.10 `ng_shutdown_t ng_hci_shutdown` [static]

Definition at line 68 of file `ng_hci_main.c`.

7.35.3.11 `ng_rcvmsg_t ng_hci_upper_rcvmsg` [static]

Definition at line 73 of file `ng_hci_main.c`.

Referenced by `ng_hci_connect()`.

7.35.3.12 `struct ng_type typestruct` [static]**Initial value:**

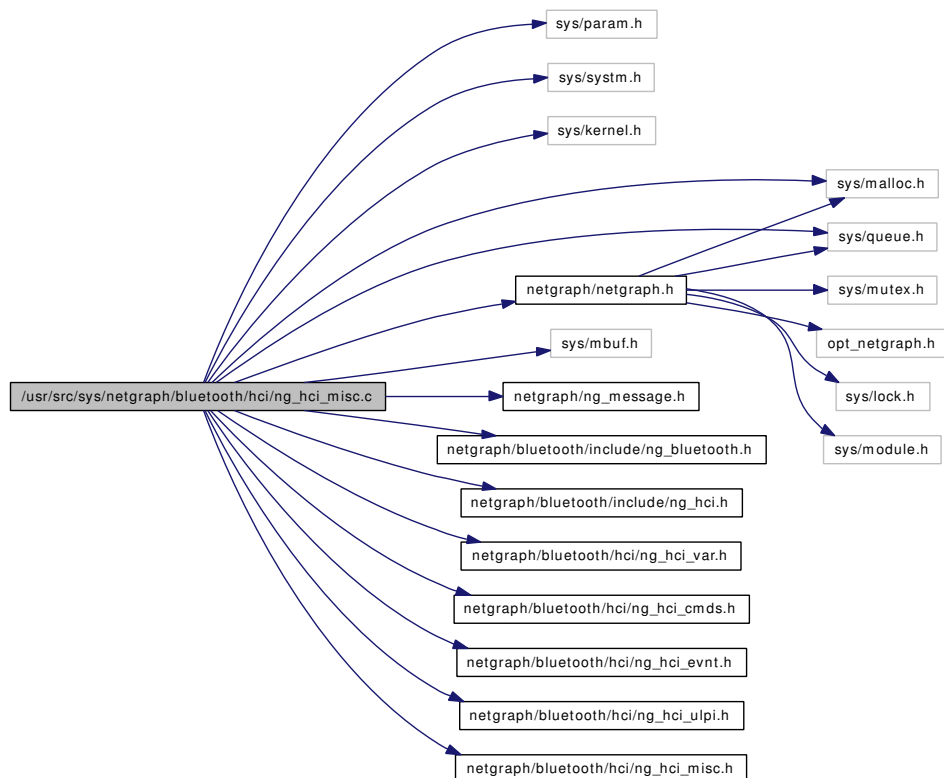
```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_HCI_NODE_TYPE,
    .constructor = ng_hci_constructor,
    .rcvmsg =       ng_hci_default_rcvmsg,
    .shutdown =     ng_hci_shutdown,
    .newhook =      ng_hci_newhook,
    .connect =      ng_hci_connect,
    .rcvdata =      ng_hci_drv_rcvdata,
    .disconnect =   ng_hci_disconnect,
    .cmdlist =      ng_hci_cmdlist,
}
```

Definition at line 80 of file `ng_hci_main.c`.

7.36 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/hci/ng_hci_var.h>
#include <netgraph/bluetooth/hci/ng_hci_cmds.h>
#include <netgraph/bluetooth/hci/ng_hci_evnt.h>
#include <netgraph/bluetooth/hci/ng_hci_ulpi.h>
#include <netgraph/bluetooth/hci/ng_hci_misc.h>
```

Include dependency graph for ng_hci_misc.c:



Functions

- void [ng_hci_mtap](#) ([ng_hci_unit_p](#) unit, struct mbuf *m0)
- void [ng_hci_node_is_up](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- void [ng_hci_unit_clean](#) ([ng_hci_unit_p](#) unit, int reason)
- [ng_hci_neighbor_p](#) [ng_hci_new_neighbor](#) ([ng_hci_unit_p](#) unit)
- void [ng_hci_free_neighbor](#) ([ng_hci_neighbor_p](#) n)
- void [ng_hci_flush_neighbor_cache](#) ([ng_hci_unit_p](#) unit)
- [ng_hci_neighbor_p](#) [ng_hci_get_neighbor](#) ([ng_hci_unit_p](#) unit, [bdaddr_p](#) bdaddr)
- int [ng_hci_neighbor_stale](#) ([ng_hci_neighbor_p](#) n)
- [ng_hci_unit_con_p](#) [ng_hci_new_con](#) ([ng_hci_unit_p](#) unit, int link_type)
- void [ng_hci_free_con](#) ([ng_hci_unit_con_p](#) con)
- [ng_hci_unit_con_p](#) [ng_hci_con_by_handle](#) ([ng_hci_unit_p](#) unit, int con_handle)
- [ng_hci_unit_con_p](#) [ng_hci_con_by_bdaddr](#) ([ng_hci_unit_p](#) unit, [bdaddr_p](#) bdaddr, int link_type)
- int [ng_hci_command_timeout](#) ([ng_hci_unit_p](#) unit)
- int [ng_hci_command_untimeout](#) ([ng_hci_unit_p](#) unit)
- int [ng_hci_con_timeout](#) ([ng_hci_unit_con_p](#) con)
- int [ng_hci_con_untimeout](#) ([ng_hci_unit_con_p](#) con)

7.36.1 Function Documentation

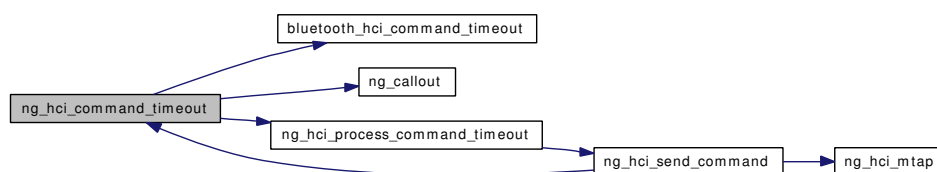
7.36.1.1 int [ng_hci_command_timeout](#) ([ng_hci_unit_p](#) unit)

Definition at line 366 of file [ng_hci_misc.c](#).

References [bluetooth_hci_command_timeout\(\)](#), [ng_hci_unit::cmd_timo](#), [ng_callout\(\)](#), [ng_hci_process_command_timeout\(\)](#), [NG_HCI_UNIT_COMMAND_PENDING](#), [NG_NODE_NAME](#), [ng_hci_unit::node](#), and [ng_hci_unit::state](#).

Referenced by [ng_hci_send_command\(\)](#).

Here is the call graph for this function:



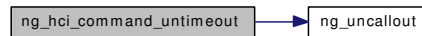
7.36.1.2 int [ng_hci_command_untimeout](#) ([ng_hci_unit_p](#) unit)

Definition at line 385 of file [ng_hci_misc.c](#).

References [ng_hci_unit::cmd_timo](#), [NG_HCI_UNIT_COMMAND_PENDING](#), [NG_NODE_NAME](#), [ng_uncallout\(\)](#), [ng_hci_unit::node](#), and [ng_hci_unit::state](#).

Referenced by [complete_command\(\)](#), and [ng_hci_unit_clean\(\)](#).

Here is the call graph for this function:



7.36.1.3 `ng_hci_unit_con_p ng_hci_con_by_bdaddr (ng_hci_unit_p unit, bdaddr_p bdaddr, int link_type)`

Definition at line 348 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::bdaddr`, and `ng_hci_unit_con::link_type`.

Referenced by `ng_hci_lp_acl_con_req()`, and `role_change()`.

7.36.1.4 `ng_hci_unit_con_p ng_hci_con_by_handle (ng_hci_unit_p unit, int con_handle)`

Definition at line 332 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::con_handle`.

Referenced by `discon_compl()`, `encryption_change()`, `mode_change()`, `ng_hci_acl_rcvdata()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_process_con_timeout()`, `ng_hci_sco_rcvdata()`, `num_compl_pkts()`, `process_link_policy_params()`, `qos_setup_compl()`, `qos_violation()`, `read_clock_offset_compl()`, and `read_remote_features_compl()`.

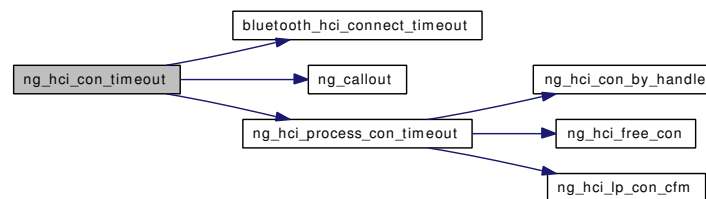
7.36.1.5 `int ng_hci_con_timeout (ng_hci_unit_con_p con)`

Definition at line 405 of file `ng_hci_misc.c`.

References `bluetooth_hci_connect_timeout()`, `ng_hci_unit_con::con_handle`, `ng_hci_unit_con::con_timo`, `ng_hci_unit_con::flags`, `ng_callout()`, `NG_HCI_CON_TIMEOUT_PENDING`, `ng_hci_process_con_timeout()`, `NG_NODE_NAME`, `ng_hci_unit::node`, and `ng_hci_unit_con::unit`.

Referenced by `con_req()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, and `ng_hci_lp_sco_con_req()`.

Here is the call graph for this function:



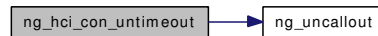
7.36.1.6 `int ng_hci_con_untimeout (ng_hci_unit_con_p con)`

Definition at line 426 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::con_timo`, `ng_hci_unit_con::flags`, `NG_HCI_CON_TIMEOUT_PENDING`, `NG_NODE_NAME`, `ng_uncallout()`, `ng_hci_unit::node`, and `ng_hci_unit_con::unit`.

Referenced by `con_compl()`, `con_req()`, `discon_compl()`, `ng_hci_lp_con_rsp()`, `ng_hci_unit_clean()`, and `process_hc_baseband_params()`.

Here is the call graph for this function:



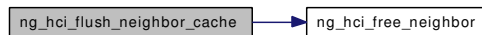
7.36.1.7 void ng_hci_flush_neighbor_cache (ng_hci_unit_p unit)

Definition at line 206 of file ng_hci_misc.c.

References ng_hci_free_neighbor().

Referenced by ng_hci_default_rcvmsg(), and ng_hci_unit_clean().

Here is the call graph for this function:



7.36.1.8 void ng_hci_free_con (ng_hci_unit_con_p con)

Definition at line 307 of file ng_hci_misc.c.

References ng_hci_unit::buffer, ng_hci_unit_con::conq, ng_hci_unit_con::link_type, M_NETGRAPH_HCI, NG_BT_ITEMQ_DESTROY, NG_HCI_BUFF_ACL_FREE, NG_HCI_BUFF_SCO_FREE, NG_HCI_LINK_ACL, ng_hci_unit_con::pending, and ng_hci_unit_con::unit.

Referenced by con_compl(), con_req(), discon_compl(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_sco_con_req(), ng_hci_process_con_timeout(), ng_hci_unit_clean(), and process_hc_baseband_params().

7.36.1.9 void ng_hci_free_neighbor (ng_hci_neighbor_p n)

Definition at line 194 of file ng_hci_misc.c.

References M_NETGRAPH_HCI.

Referenced by ng_hci_default_rcvmsg(), ng_hci_flush_neighbor_cache(), and ng_hci_get_neighbor().

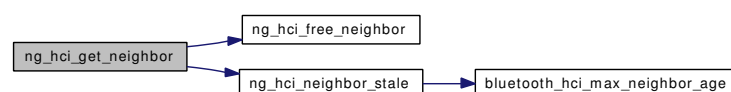
7.36.1.10 ng_hci_neighbor_p ng_hci_get_neighbor (ng_hci_unit_p unit, bdaddr_p bdaddr)

Definition at line 217 of file ng_hci_misc.c.

References ng_hci_neighbor::bdaddr, ng_hci_free_neighbor(), and ng_hci_neighbor_stale().

Referenced by inquiry_result(), ng_hci_lp_acl_con_req(), page_scan_mode_change(), page_scan_rep_mode_change(), read_clock_offset_compl(), and read_remote_features_compl().

Here is the call graph for this function:



7.36.1.11 void ng_hci_mtap (ng_hci_unit_p unit, struct mbuf * m0)

Definition at line 62 of file ng_hci_misc.c.

References NG_HCI_INFO, NG_HOOK_IS_VALID, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_hci_unit::node, and ng_hci_unit::raw.

Referenced by ng_hci_drv_rcvdata(), ng_hci_send_command(), and send_data_packets().

7.36.1.12 int ng_hci_neighbor_stale (ng_hci_neighbor_p n)

Definition at line 241 of file ng_hci_misc.c.

References bluetooth_hci_max_neighbor_age(), and ng_hci_neighbor::updated.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_get_neighbor().

Here is the call graph for this function:

**7.36.1.13 ng_hci_unit_con_p ng_hci_new_con (ng_hci_unit_p unit, int link_type)**

Definition at line 255 of file ng_hci_misc.c.

References ng_hci_unit::buffer, M_NETGRAPH_HCI, NG_BT_ITEMQ_INIT, ng_callout_init, NG_HCI_BUFF_ACL_TOTAL, NG_HCI_BUFF_SCO_TOTAL, NG_HCI_CON_CLOSED, NG_HCI_LINK_ACL, and ng_hci_unit::state.

Referenced by con_compl(), con_req(), ng_hci_lp_acl_con_req(), and ng_hci_lp_sco_con_req().

7.36.1.14 ng_hci_neighbor_p ng_hci_new_neighbor (ng_hci_unit_p unit)

Definition at line 175 of file ng_hci_misc.c.

References M_NETGRAPH_HCI.

Referenced by inquiry_result(), page_scan_mode_change(), page_scan_rep_mode_change(), read_clock_offset_compl(), and read_remote_features_compl().

7.36.1.15 void ng_hci_node_is_up (node_p node, hook_p hook, void * arg1, int arg2)

Definition at line 84 of file ng_hci_misc.c.

References ng_hci_unit::acl, ng_hci_unit::bdaddr, ng_hci_unit::buffer, ng_mesg::data, NG_HCI_BUFF_ACL_SIZE, NG_HCI_BUFF_ACL_TOTAL, NG_HCI_BUFF_SCO_SIZE, NG_HCI_BUFF_SCO_TOTAL, NG_HCI_INFO, NG_HCI_UNIT_READY, NG_HOOK_NAME, NG_HOOK_NOT_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_NODE_UP, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit::state.

Referenced by ng_hci_connect(), ng_hci_default_rcvmsg(), and process_info_params().

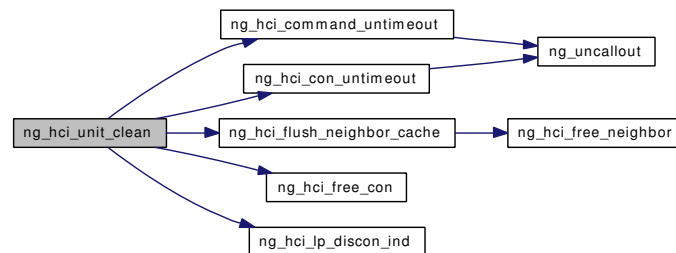
7.36.1.16 void ng_hci_unit_clean (ng_hci_unit_p unit, int reason)

Definition at line 132 of file ng_hci_misc.c.

References `ng_hci_unit_con::flags`, `NG_BT_MBUFQ_DRAIN`, `NG_HCI_BUFF_ACL_FREE`, `NG_HCI_BUFF_ACL_TOTAL`, `NG_HCI_BUFF_CMD_SET`, `NG_HCI_BUFF_SCO_FREE`, `NG_HCI_BUFF_SCO_TOTAL`, `ng_hci_command_untimeout()`, `NG_HCI_CON_TIMEOUT_PENDING`, `ng_hci_con_untimeout()`, `ng_hci_flush_neighbor_cache()`, `ng_hci_free_con()`, `ng_hci_lp_discon_ind()`, `NG_HCI_UNIT_COMMAND_PENDING`, and `ng_hci_unit::state`.

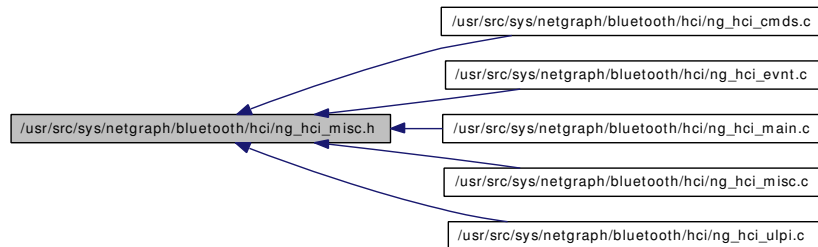
Referenced by `ng_hci_disconnect()`, and `ng_hci_shutdown()`.

Here is the call graph for this function:



7.37 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [ng_hci_mtap](#) ([ng_hci_unit_p](#), struct mbuf *)
- void [ng_hci_node_is_up](#) ([node_p](#), [hook_p](#), void *, int)
- void [ng_hci_unit_clean](#) ([ng_hci_unit_p](#), int)
- [ng_hci_neighbor_p](#) [ng_hci_new_neighbor](#) ([ng_hci_unit_p](#))
- void [ng_hci_free_neighbor](#) ([ng_hci_neighbor_p](#))
- void [ng_hci_flush_neighbor_cache](#) ([ng_hci_unit_p](#))
- [ng_hci_neighbor_p](#) [ng_hci_get_neighbor](#) ([ng_hci_unit_p](#), [bdaddr_p](#))
- int [ng_hci_neighbor_stale](#) ([ng_hci_neighbor_p](#))
- [ng_hci_unit_con_p](#) [ng_hci_new_con](#) ([ng_hci_unit_p](#), int)
- void [ng_hci_free_con](#) ([ng_hci_unit_con_p](#))
- [ng_hci_unit_con_p](#) [ng_hci_con_by_handle](#) ([ng_hci_unit_p](#), int)
- [ng_hci_unit_con_p](#) [ng_hci_con_by_bdaddr](#) ([ng_hci_unit_p](#), [bdaddr_p](#), int)
- int [ng_hci_command_timeout](#) ([ng_hci_unit_p](#))
- int [ng_hci_command_untimeout](#) ([ng_hci_unit_p](#))
- int [ng_hci_con_timeout](#) ([ng_hci_unit_con_p](#))
- int [ng_hci_con_untimeout](#) ([ng_hci_unit_con_p](#))

7.37.1 Function Documentation

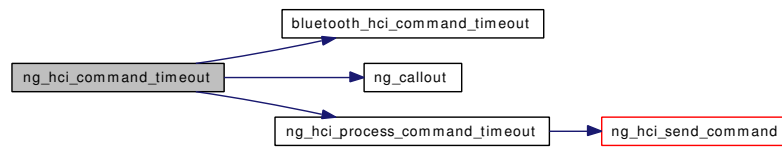
7.37.1.1 int [ng_hci_command_timeout](#) ([ng_hci_unit_p](#))

Definition at line 366 of file [ng_hci_misc.c](#).

References [bluetooth_hci_command_timeout\(\)](#), [ng_hci_unit::cmd_timo](#), [ng_callout\(\)](#), [ng_hci_process_command_timeout\(\)](#), [NG_HCI_UNIT_COMMAND_PENDING](#), [NG_NODE_NAME](#), [ng_hci_unit::node](#), and [ng_hci_unit::state](#).

Referenced by [ng_hci_send_command\(\)](#).

Here is the call graph for this function:



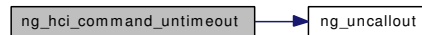
7.37.1.2 `int ng_hci_command_untimeout (ng_hci_unit_p)`

Definition at line 385 of file `ng_hci_misc.c`.

References `ng_hci_unit::cmd_timo`, `NG_HCI_UNIT_COMMAND_PENDING`, `NG_NODE_NAME`, `ng_uncallout()`, `ng_hci_unit::node`, and `ng_hci_unit::state`.

Referenced by `complete_command()`, and `ng_hci_unit_clean()`.

Here is the call graph for this function:



7.37.1.3 `ng_hci_unit_con_p ng_hci_con_by_bdaddr (ng_hci_unit_p, bdaddr_p, int)`

Definition at line 348 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::bdaddr`, and `ng_hci_unit_con::link_type`.

Referenced by `ng_hci_lp_acl_con_req()`, and `role_change()`.

7.37.1.4 `ng_hci_unit_con_p ng_hci_con_by_handle (ng_hci_unit_p, int)`

Definition at line 332 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::con_handle`.

Referenced by `discon_compl()`, `encryption_change()`, `mode_change()`, `ng_hci_acl_rcvdata()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_process_con_timeout()`, `ng_hci_sco_rcvdata()`, `num_compl_pkts()`, `process_link_policy_params()`, `qos_setup_compl()`, `qos_violation()`, `read_clock_offset_compl()`, and `read_remote_features_compl()`.

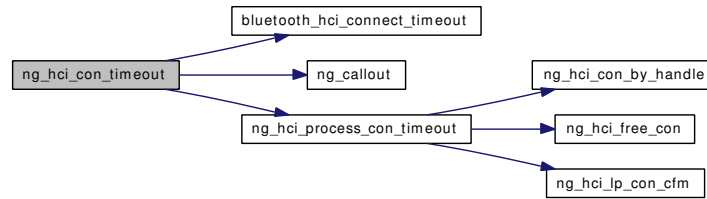
7.37.1.5 `int ng_hci_con_timeout (ng_hci_unit_con_p)`

Definition at line 405 of file `ng_hci_misc.c`.

References `bluetooth_hci_connect_timeout()`, `ng_hci_unit_con::con_handle`, `ng_hci_unit_con::con_timo`, `ng_hci_unit_con::flags`, `ng_callout()`, `NG_HCI_CON_TIMEOUT_PENDING`, `ng_hci_process_con_timeout()`, `NG_NODE_NAME`, `ng_hci_unit::node`, and `ng_hci_unit_con::unit`.

Referenced by `con_req()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, and `ng_hci_lp_sco_con_req()`.

Here is the call graph for this function:



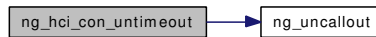
7.37.1.6 int ng_hci_con_untimeout (ng_hci_unit_con_p)

Definition at line 426 of file ng_hci_misc.c.

References ng_hci_unit_con::con_timo, ng_hci_unit_con::flags, NG_HCI_CON_TIMEOUT_PENDING, NG_NODE_NAME, ng_uncallout(), ng_hci_unit::node, and ng_hci_unit_con::unit.

Referenced by con_compl(), con_req(), discon_compl(), ng_hci_lp_con_rsp(), ng_hci_unit_clean(), and process_hc_baseband_params().

Here is the call graph for this function:



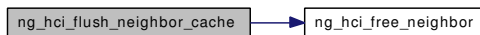
7.37.1.7 void ng_hci_flush_neighbor_cache (ng_hci_unit_p)

Definition at line 206 of file ng_hci_misc.c.

References ng_hci_free_neighbor().

Referenced by ng_hci_default_rcvmsg(), and ng_hci_unit_clean().

Here is the call graph for this function:



7.37.1.8 void ng_hci_free_con (ng_hci_unit_con_p)

Definition at line 307 of file ng_hci_misc.c.

References ng_hci_unit::buffer, ng_hci_unit_con::conq, ng_hci_unit_con::link_type, M_NETGRAPH_HCI, NG_BT_ITEMQ_DESTROY, NG_HCI_BUFF_ACL_FREE, NG_HCI_BUFF_SCO_FREE, NG_HCI_LINK_ACL, ng_hci_unit_con::pending, and ng_hci_unit_con::unit.

Referenced by con_compl(), con_req(), discon_compl(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_sco_con_req(), ng_hci_process_con_timeout(), ng_hci_unit_clean(), and process_hc_baseband_params().

7.37.1.9 void ng_hci_free_neighbor (ng_hci_neighbor_p)

Definition at line 194 of file ng_hci_misc.c.

References M_NETGRAPH_HCI.

Referenced by ng_hci_default_rcvmsg(), ng_hci_flush_neighbor_cache(), and ng_hci_get_neighbor().

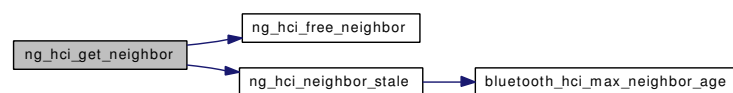
7.37.1.10 `ng_hci_neighbor_p` `ng_hci_get_neighbor` (`ng_hci_unit_p`, `bdaddr_p`)

Definition at line 217 of file ng_hci_misc.c.

References ng_hci_neighbor::bdaddr, ng_hci_free_neighbor(), and ng_hci_neighbor_stale().

Referenced by inquiry_result(), ng_hci_lp_acl_con_req(), page_scan_mode_change(), page_scan_rep_mode_change(), read_clock_offset_compl(), and read_remote_features_compl().

Here is the call graph for this function:



7.37.1.11 `void` `ng_hci_mtap` (`ng_hci_unit_p`, `struct mbuf *`)

Definition at line 62 of file ng_hci_misc.c.

References NG_HCI_INFO, NG_HOOK_IS_VALID, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_hci_unit::node, and ng_hci_unit::raw.

Referenced by ng_hci_drv_rcvdata(), ng_hci_send_command(), and send_data_packets().

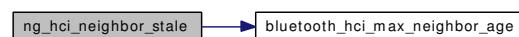
7.37.1.12 `int` `ng_hci_neighbor_stale` (`ng_hci_neighbor_p`)

Definition at line 241 of file ng_hci_misc.c.

References bluetooth_hci_max_neighbor_age(), and ng_hci_neighbor::updated.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_get_neighbor().

Here is the call graph for this function:



7.37.1.13 `ng_hci_unit_con_p` `ng_hci_new_con` (`ng_hci_unit_p`, `int`)

Definition at line 255 of file ng_hci_misc.c.

References ng_hci_unit::buffer, M_NETGRAPH_HCI, NG_BT_ITEMQ_INIT, ng_callout_init, NG_HCI_BUFF_ACL_TOTAL, NG_HCI_BUFF_SCO_TOTAL, NG_HCI_CON_CLOSED, NG_HCI_LINK_ACL, and ng_hci_unit::state.

Referenced by con_compl(), con_req(), ng_hci_lp_acl_con_req(), and ng_hci_lp_sco_con_req().

7.37.1.14 `ng_hci_neighbor_p ng_hci_new_neighbor (ng_hci_unit_p)`

Definition at line 175 of file `ng_hci_misc.c`.

References `M_NETGRAPH_HCI`.

Referenced by `inquiry_result()`, `page_scan_mode_change()`, `page_scan_rep_mode_change()`, `read_clock_offset_compl()`, and `read_remote_features_compl()`.

7.37.1.15 `void ng_hci_node_is_up (node_p, hook_p, void *, int)`

Definition at line 84 of file `ng_hci_misc.c`.

References `ng_hci_unit::acl`, `ng_hci_unit::bdaddr`, `ng_hci_unit::buffer`, `ng_mesg::data`, `NG_HCI_BUFF_ACL_SIZE`, `NG_HCI_BUFF_ACL_TOTAL`, `NG_HCI_BUFF_SCO_SIZE`, `NG_HCI_BUFF_SCO_TOTAL`, `NG_HCI_INFO`, `NG_HCI_UNIT_READY`, `NG_HOOK_NAME`, `NG_HOOK_NOT_VALID`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_NODE_UP`, `ng_hci_unit::node`, `ng_hci_unit::sco`, and `ng_hci_unit::state`.

Referenced by `ng_hci_connect()`, `ng_hci_default_rcvmsg()`, and `process_info_params()`.

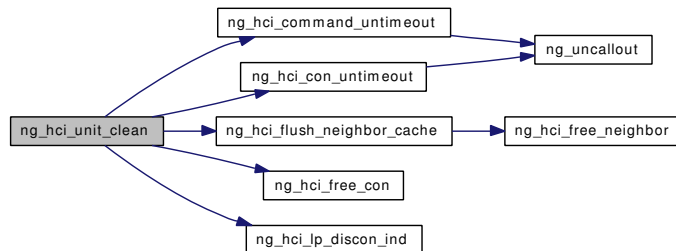
7.37.1.16 `void ng_hci_unit_clean (ng_hci_unit_p, int)`

Definition at line 132 of file `ng_hci_misc.c`.

References `ng_hci_unit_con::flags`, `NG_BT_MBUFQ_DRAIN`, `NG_HCI_BUFF_ACL_FREE`, `NG_HCI_BUFF_ACL_TOTAL`, `NG_HCI_BUFF_CMD_SET`, `NG_HCI_BUFF_SCO_FREE`, `NG_HCI_BUFF_SCO_TOTAL`, `ng_hci_command_untimeout()`, `NG_HCI_CON_TIMEOUT_PENDING`, `ng_hci_con_untimeout()`, `ng_hci_flush_neighbor_cache()`, `ng_hci_free_con()`, `ng_hci_lp_discon_ind()`, `NG_HCI_UNIT_COMMAND_PENDING`, and `ng_hci_unit::state`.

Referenced by `ng_hci_disconnect()`, and `ng_hci_shutdown()`.

Here is the call graph for this function:



7.38 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_prse.h File Reference

This graph shows which files directly or indirectly include this file:



Variables

- static struct [ng_parse_fixedarray_info](#) [ng_hci_bdaddr_type_info](#)
- static struct [ng_parse_type](#) [ng_hci_bdaddr_type](#)
- static struct [ng_parse_fixedarray_info](#) [ng_hci_features_type_info](#)
- static struct [ng_parse_type](#) [ng_hci_features_type](#)
- static struct [ng_parse_struct_field](#) [ng_hci_buffer_type_fields](#) []
- static struct [ng_parse_type](#) [ng_hci_buffer_type](#)
- static struct [ng_parse_struct_field](#) [ng_hci_stat_type_fields](#) []
- static struct [ng_parse_type](#) [ng_hci_stat_type](#)
- static struct [ng_cmdlist](#) [ng_hci_cmdlist](#) []

7.38.1 Variable Documentation

7.38.1.1 struct [ng_parse_type](#) [ng_hci_bdaddr_type](#) [static]

Initial value:

```

{
    &ng_parse_fixedarray_type,
    &ng_hci_bdaddr_type_info
}
  
```

Definition at line 48 of file [ng_hci_prse.h](#).

7.38.1.2 struct [ng_parse_fixedarray_info](#) [ng_hci_bdaddr_type_info](#) [static]

Initial value:

```

{
    &ng_parse_uint8_type,
    NG_HCI_BDADDR_SIZE
}
  
```

Definition at line 44 of file [ng_hci_prse.h](#).

7.38.1.3 struct [ng_parse_type](#) [ng_hci_buffer_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_hci_buffer_type_fields
}
```

Definition at line 75 of file ng_hci_prse.h.

7.38.1.4 struct [ng_parse_struct_field ng_hci_buffer_type_fields\[\]](#) [static]

Initial value:

```
{
    { "cmd_free",    &ng_parse_uint8_type,  },
    { "sco_size",   &ng_parse_uint8_type,  },
    { "sco_pkts",   &ng_parse_uint16_type, },
    { "sco_free",   &ng_parse_uint16_type, },
    { "acl_size",   &ng_parse_uint16_type, },
    { "acl_pkts",   &ng_parse_uint16_type, },
    { "acl_free",   &ng_parse_uint16_type, },
    { NULL,        }
}
```

Definition at line 64 of file ng_hci_prse.h.

7.38.1.5 struct [ng_cmdlist ng_hci_cmdlist\[\]](#) [static]

Definition at line 102 of file ng_hci_prse.h.

7.38.1.6 struct [ng_parse_type ng_hci_features_type](#) [static]

Initial value:

```
{
    &ng_parse_fixedarray_type,
    &ng_hci_features_type_info
}
```

Definition at line 58 of file ng_hci_prse.h.

7.38.1.7 struct [ng_parse_fixedarray_info ng_hci_features_type_info](#) [static]

Initial value:

```
{
    &ng_parse_uint8_type,
    NG_HCI_FEATURES_SIZE
}
```

Definition at line 54 of file ng_hci_prse.h.

7.38.1.8 struct `ng_parse_type ng_hci_stat_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_hci_stat_type_fields
}
```

Definition at line 93 of file `ng_hci_prse.h`.

7.38.1.9 struct `ng_parse_struct_field ng_hci_stat_type_fields[]` [static]**Initial value:**

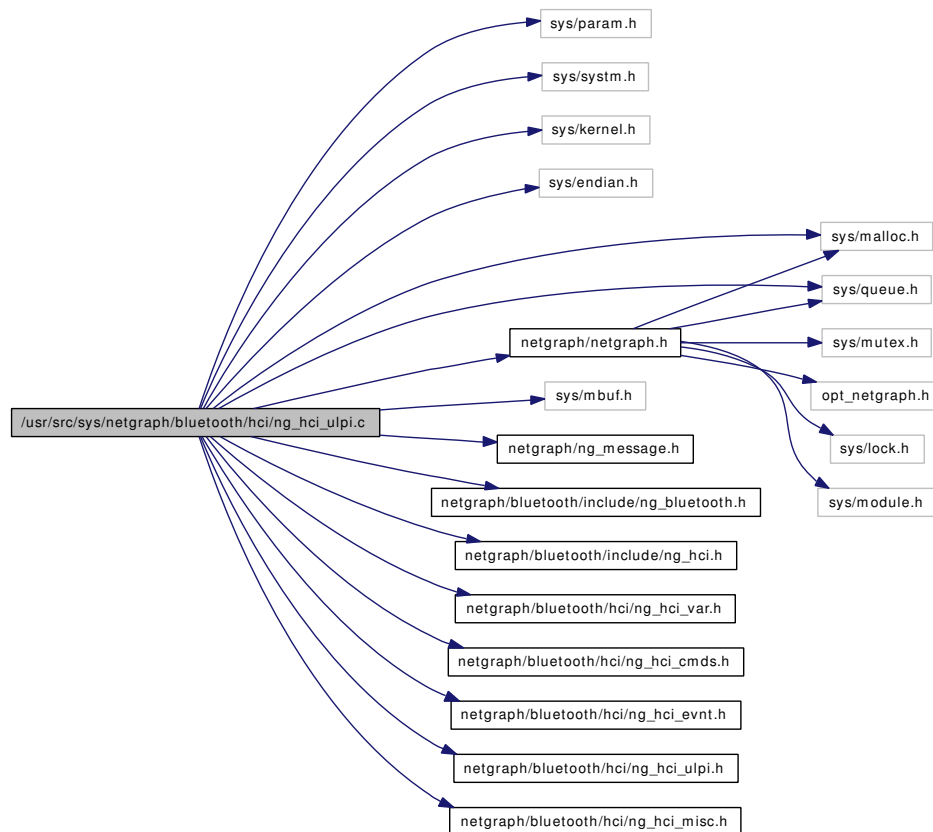
```
{
    { "cmd_sent",    &ng_parse_uint32_type, },
    { "evnt_recv",  &ng_parse_uint32_type, },
    { "acl_recv",   &ng_parse_uint32_type, },
    { "acl_sent",   &ng_parse_uint32_type, },
    { "sco_recv",   &ng_parse_uint32_type, },
    { "sco_sent",   &ng_parse_uint32_type, },
    { "bytes_recv", &ng_parse_uint32_type, },
    { "bytes_sent", &ng_parse_uint32_type, },
    { NULL, }
}
```

Definition at line 81 of file `ng_hci_prse.h`.

7.39 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/hci/ng_hci_var.h>
#include <netgraph/bluetooth/hci/ng_hci_cmds.h>
#include <netgraph/bluetooth/hci/ng_hci_evnt.h>
#include <netgraph/bluetooth/hci/ng_hci_ulpi.h>
#include <netgraph/bluetooth/hci/ng_hci_misc.h>
```

Include dependency graph for ng_hci_ulpi.c:



Functions

- static int [ng_hci_lp_acl_con_req](#) ([ng_hci_unit_p](#), [item_p](#), [hook_p](#))
- static int [ng_hci_lp_sco_con_req](#) ([ng_hci_unit_p](#), [item_p](#), [hook_p](#))
- int [ng_hci_lp_con_req](#) ([ng_hci_unit_p](#) unit, [item_p](#) item, [hook_p](#) hook)
- int [ng_hci_lp_discon_req](#) ([ng_hci_unit_p](#) unit, [item_p](#) item, [hook_p](#) hook)
- int [ng_hci_lp_con_cfm](#) ([ng_hci_unit_con_p](#) con, int status)
- int [ng_hci_lp_con_ind](#) ([ng_hci_unit_con_p](#) con, u_int8_t *uclass)
- int [ng_hci_lp_con_rsp](#) ([ng_hci_unit_p](#) unit, [item_p](#) item, [hook_p](#) hook)
- int [ng_hci_lp_discon_ind](#) ([ng_hci_unit_con_p](#) con, int reason)
- int [ng_hci_lp_qos_req](#) ([ng_hci_unit_p](#) unit, [item_p](#) item, [hook_p](#) hook)
- int [ng_hci_lp_qos_cfm](#) ([ng_hci_unit_con_p](#) con, int status)
- int [ng_hci_lp_qos_ind](#) ([ng_hci_unit_con_p](#) con)
- void [ng_hci_process_con_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int con_handle)

7.39.1 Function Documentation

7.39.1.1 static int [ng_hci_lp_acl_con_req](#) ([ng_hci_unit_p](#), [item_p](#), [hook_p](#)) [static]

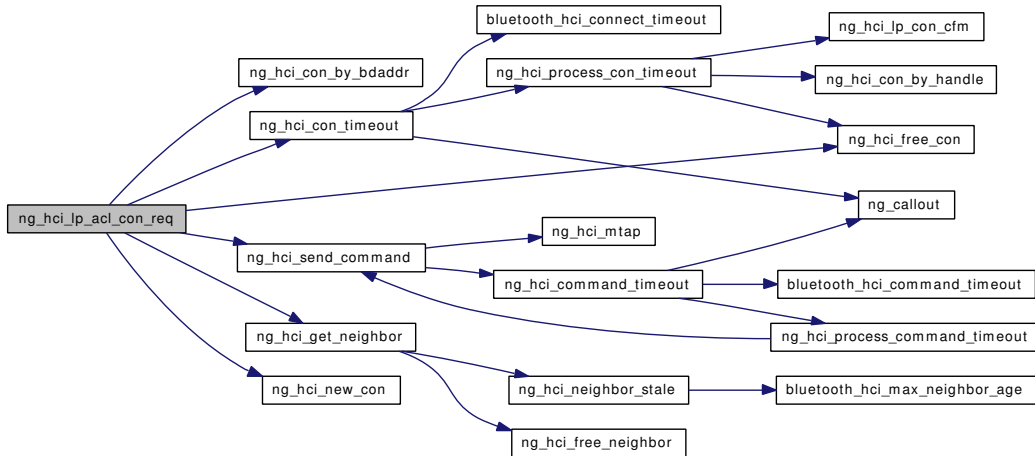
Definition at line 109 of file [ng_hci_ulpi.c](#).

References [_NGI_MSG](#), [ng_hci_unit::acl](#), [ng_hci_unit_con::bdaddr](#), [ng_hci_lp_con_req_ep::bdaddr](#), [ng_hci_neighbor::clock_offset](#), [ng_hci_unit::cmdq](#), [ng_hci_unit_con::con_handle](#), [ng_mesg::data](#), [ng_hci_unit::features](#), [ng_hci_unit_con::flags](#), [ng_hci_unit_con::link_type](#), [NG_BT_MBUFQ_ENQUEUE](#),

NG_FREE_ITEM, NG_FREE_MSG, NG_FWD_ITEM_HOOK, NG_HCI_CMD_PKT, ng_hci_con_by_bdaddr(), NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, NG_HCI_CON_OPEN, ng_hci_con_timeout(), NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), ng_hci_get_neighbor(), NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HCI_LMP_3SLOT, NG_HCI_LMP_5SLOT, NG_HCI_LMP_SWITCH, ng_hci_new_con(), NG_HCI_OCF_CREATE_CON, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OPCODE, NG_HCI_PKT_DH1, NG_HCI_PKT_DH3, NG_HCI_PKT_DH5, NG_HCI_PKT_DM1, NG_HCI_PKT_DM3, NG_HCI_PKT_DM5, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NGI_GET_MSG, NGI_MSG, NGM_HCI_COOKIE, NGM_HCI_LP_CON_CFM, ng_hci_unit::node, packed, ng_hci_unit::packet_mask, ng_hci_neighbor::page_scan_mode, ng_hci_neighbor::page_scan_rep_mode, ng_hci_unit::role_switch, ng_hci_unit::state, and ng_hci_unit_con::state.

Referenced by ng_hci_lp_con_req().

Here is the call graph for this function:



7.39.1.2 int ng_hci_lp_con_cfm (ng_hci_unit_con_p con, int status)

Definition at line 568 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::bdaddr, ng_hci_unit_con::con_handle, ng_mesg::data, ng_hci_unit_con::flags, ng_hci_unit_con::link_type, NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_CON_CFM, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by con_compl(), and ng_hci_process_con_timeout().

7.39.1.3 int ng_hci_lp_con_ind (ng_hci_unit_con_p con, u_int8_t * uclass)

Definition at line 636 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::bdaddr, ng_hci_unit_con::link_type, NG_HCI_LINK_ACL, NG_HCI_WARN, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_CON_IND, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by `con_req()`.

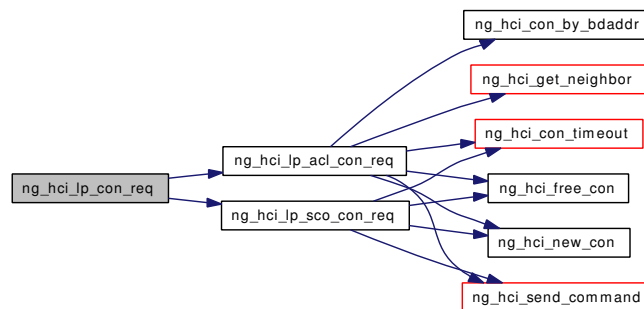
7.39.1.4 `int ng_hci_lp_con_req (ng_hci_unit_p unit, item_p item, hook_p hook)`

Definition at line 65 of file `ng_hci_ulpi.c`.

References `NG_FREE_ITEM`, `NG_HCI_ALERT`, `NG_HCI_LINK_ACL`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_sco_con_req()`, `NG_HCI_UNIT_READY`, `NG_HCI_WARN`, `NG_NODE_NAME`, `NGI_MSG`, `ng_hci_unit::node`, and `ng_hci_unit::state`.

Referenced by `ng_hci_upper_rcvmsg()`.

Here is the call graph for this function:



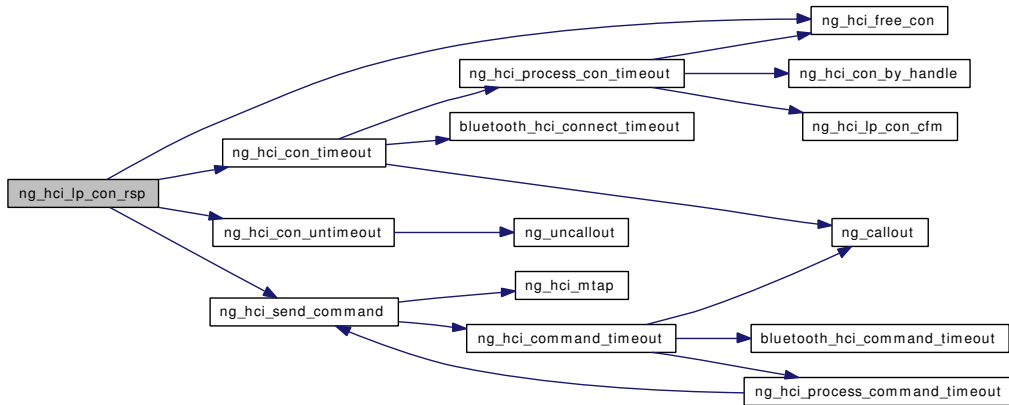
7.39.1.5 `int ng_hci_lp_con_rsp (ng_hci_unit_p unit, item_p item, hook_p hook)`

Definition at line 682 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::bdaddr`, `ng_hci_unit::cmdq`, `ng_hci_unit::features`, `ng_hci_unit_con::flags`, `ng_hci_unit_con::link_type`, `NG_BT_MBUFQ_ENQUEUE`, `NG_FREE_ITEM`, `NG_HCI_ALERT`, `NG_HCI_CMD_PKT`, `NG_HCI_CON_NOTIFY_ACL`, `NG_HCI_CON_NOTIFY_SCO`, `ng_hci_con_timeout()`, `ng_hci_con_untimeout()`, `NG_HCI_CON_W4_CONN_COMPLETE`, `NG_HCI_CON_W4_LP_CON_RSP`, `ng_hci_free_con()`, `NG_HCI_LMP_SWITCH`, `NG_HCI_OCF_ACCEPT_CON`, `NG_HCI_OCF_REJECT_CON`, `NG_HCI_OGF_LINK_CONTROL`, `NG_HCI_OPCODE`, `NG_HCI_ROLE_MASTER`, `NG_HCI_ROLE_SLAVE`, `ng_hci_send_command()`, `NG_HCI_UNIT_COMMAND_PENDING`, `NG_HCI_UNIT_READY`, `NG_HCI_WARN`, `NG_NODE_NAME`, `NGI_MSG`, `ng_hci_unit::node`, `packed`, `ng_hci_unit::role_switch`, `ng_hci_unit::state`, and `ng_hci_unit_con::state`.

Referenced by `ng_hci_upper_rcvmsg()`.

Here is the call graph for this function:



7.39.1.6 int ng_hci_lp_discon_ind (ng_hci_unit_con_p con, int reason)

Definition at line 876 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::con_handle, ng_mesg::data, ng_hci_unit_con::link_type, NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_DISCON_IND, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by discon_compl(), ng_hci_unit_clean(), and process_hc_baseband_params().

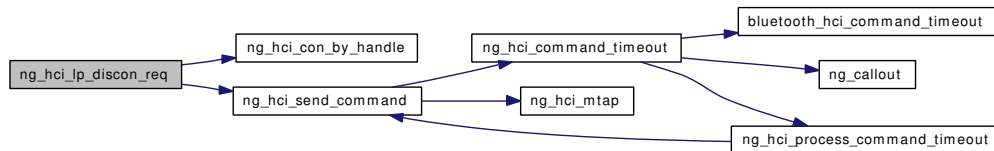
7.39.1.7 int ng_hci_lp_discon_req (ng_hci_unit_p unit, item_p item, hook_p hook)

Definition at line 477 of file ng_hci_ulpi.c.

References ng_hci_unit::cmdq, NG_BT_MBUFQ_ENQUEUE, NG_FREE_ITEM, NG_HCI_ALERT, NG_HCI_CMD_PKT, ng_hci_con_by_handle(), NG_HCI_CON_OPEN, NG_HCI_ERR, NG_HCI_OCF_DISCON, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OPCODE, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_HCI_UNIT_READY, NG_HCI_WARN, NG_NODE_NAME, NGL_MSG, ng_hci_unit::node, packed, ng_hci_unit::state, and ng_hci_unit_con::state.

Referenced by ng_hci_upper_rcvmsg().

Here is the call graph for this function:



7.39.1.8 int ng_hci_lp_qos_cfm (ng_hci_unit_con_p con, int status)

Definition at line 1047 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::con_handle, ng_mesg::data, ng_hci_unit_con::flags, NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, NG_HCI_INFO, NG_HOOK_IS_VALID,

NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_QOS_CFM, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by qos_setup_compl().

7.39.1.9 int ng_hci_lp_qos_ind (ng_hci_unit_con_p con)

Definition at line 1102 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::con_handle, ng_mesg::data, NG_HCI_INFO, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_QOS_IND, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by qos_violation().

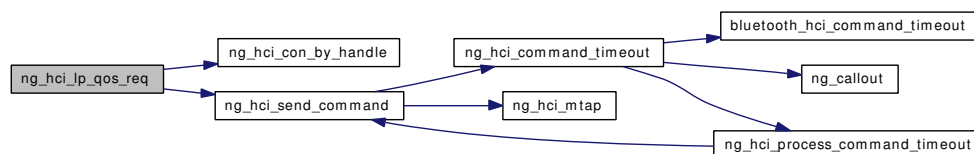
7.39.1.10 int ng_hci_lp_qos_req (ng_hci_unit_p unit, item_p item, hook_p hook)

Definition at line 934 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit::cmdq, ng_hci_unit_con::con_handle, ng_hci_unit_con::flags, ng_hci_unit_con::link_type, NG_BT_MBUFQ_ENQUEUE, NG_FREE_ITEM, NG_HCI_ALERT, NG_HCI_CMD_PKT, ng_hci_con_by_handle(), NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, NG_HCI_CON_OPEN, NG_HCI_ERR, NG_HCI_LINK_ACL, NG_HCI_OCF_QOS_SETUP, NG_HCI_OGF_LINK_POLICY, NG_HCI_OPCODE, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_HCI_UNIT_READY, NG_HCI_WARN, NG_NODE_NAME, NGI_MSG, ng_hci_unit::node, packed, ng_hci_unit::state, and ng_hci_unit_con::state.

Referenced by ng_hci_upper_rcvmsg().

Here is the call graph for this function:



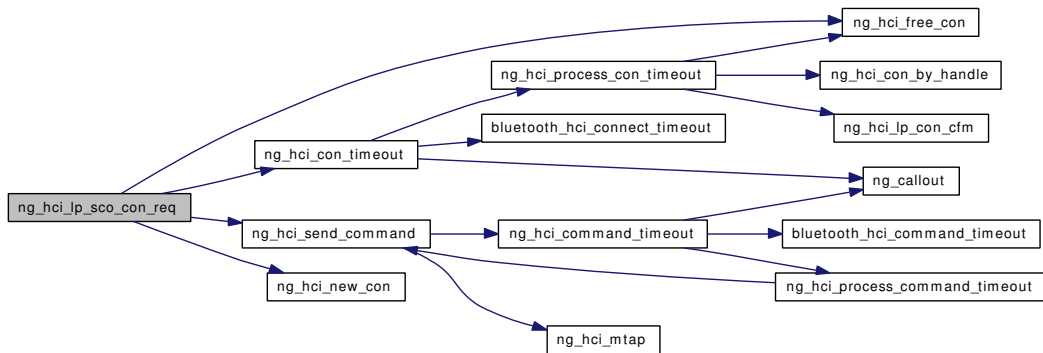
7.39.1.11 static int ng_hci_lp_sco_con_req (ng_hci_unit_p, item_p, hook_p) [static]

Definition at line 309 of file ng_hci_ulpi.c.

References ng_hci_lp_con_req_ep::bdaddr, ng_hci_unit_con::bdaddr, ng_hci_unit::cmdq, ng_hci_unit_con::con_handle, ng_hci_unit::features, ng_hci_unit_con::link_type, NG_BT_MBUFQ_ENQUEUE, NG_FREE_ITEM, NG_HCI_CMD_PKT, NG_HCI_CON_NOTIFY_SCO, NG_HCI_CON_OPEN, ng_hci_con_timeout(), NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), NG_HCI_INFO, NG_HCI_LINK_ACL, NG_HCI_LINK_SCO, NG_HCI_LMP_HV2_PKT, NG_HCI_LMP_HV3_PKT, ng_hci_new_con(), NG_HCI_OCF_ADD_SCO_CON, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OPCODE, NG_HCI_PKT_HV1, NG_HCI_PKT_HV2, NG_HCI_PKT_HV3, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_NODE_NAME, NGI_MSG, ng_hci_unit::node, packed, ng_hci_unit::packet_mask, ng_hci_unit::state, and ng_hci_unit_con::state.

Referenced by ng_hci_lp_con_req().

Here is the call graph for this function:



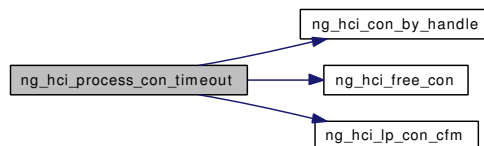
7.39.1.12 void ng_hci_process_con_timeout (node_p node, hook_p hook, void * arg1, int con_handle)

Definition at line 1152 of file ng_hci_ulpi.c.

References ng_hci_unit_con::flags, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_TIMEOUT_PENDING, NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), ng_hci_lp_con_cfm(), NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, and ng_hci_unit_con::state.

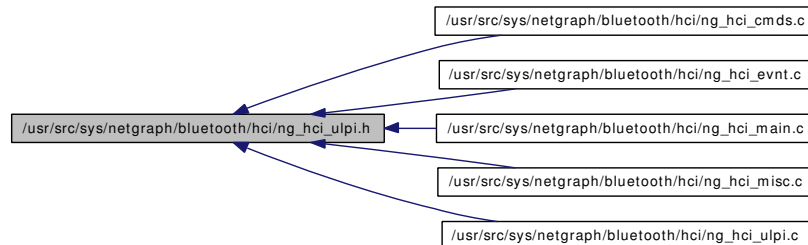
Referenced by ng_hci_con_timeout().

Here is the call graph for this function:



7.40 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int ng_hci_lp_con_req (ng_hci_unit_p, item_p, hook_p)`
- `int ng_hci_lp_discon_req (ng_hci_unit_p, item_p, hook_p)`
- `int ng_hci_lp_con_cfm (ng_hci_unit_con_p, int)`
- `int ng_hci_lp_con_ind (ng_hci_unit_con_p, u_int8_t *)`
- `int ng_hci_lp_con_rsp (ng_hci_unit_p, item_p, hook_p)`
- `int ng_hci_lp_discon_ind (ng_hci_unit_con_p, int)`
- `int ng_hci_lp_qos_req (ng_hci_unit_p, item_p, hook_p)`
- `int ng_hci_lp_qos_cfm (ng_hci_unit_con_p, int)`
- `int ng_hci_lp_qos_ind (ng_hci_unit_con_p)`
- `void ng_hci_process_con_timeout (node_p, hook_p, void *, int)`

7.40.1 Function Documentation

7.40.1.1 `int ng_hci_lp_con_cfm (ng_hci_unit_con_p, int)`

Definition at line 568 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::bdaddr`, `ng_hci_unit_con::con_handle`, `ng_mesg::data`, `ng_hci_unit_con::flags`, `ng_hci_unit_con::link_type`, `NG_HCI_CON_NOTIFY_ACL`, `NG_HCI_CON_NOTIFY_SCO`, `NG_HCI_INFO`, `NG_HCI_LINK_ACL`, `NG_HOOK_IS_VALID`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_CON_CFM`, `ng_hci_unit::node`, `ng_hci_unit::sco`, and `ng_hci_unit_con::unit`.

Referenced by `con_compl()`, and `ng_hci_process_con_timeout()`.

7.40.1.2 `int ng_hci_lp_con_ind (ng_hci_unit_con_p, u_int8_t *)`

Definition at line 636 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::bdaddr`, `ng_hci_unit_con::link_type`, `NG_HCI_LINK_ACL`, `NG_HCI_WARN`, `NG_HOOK_IS_VALID`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_CON_IND`, `ng_hci_unit::node`, `ng_hci_unit::sco`, and `ng_hci_unit_con::unit`.

Referenced by `con_req()`.

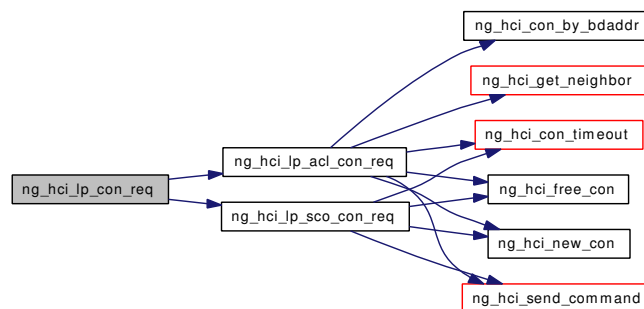
7.40.1.3 int ng_hci_lp_con_req (ng_hci_unit_p, item_p, hook_p)

Definition at line 65 of file ng_hci_ulpi.c.

References NG_FREE_ITEM, NG_HCI_ALERT, NG_HCI_LINK_ACL, ng_hci_lp_acl_con_req(), ng_hci_lp_sco_con_req(), NG_HCI_UNIT_READY, NG_HCI_WARN, NG_NODE_NAME, NGI_MSG, ng_hci_unit::node, and ng_hci_unit::state.

Referenced by ng_hci_upper_rcvmsg().

Here is the call graph for this function:



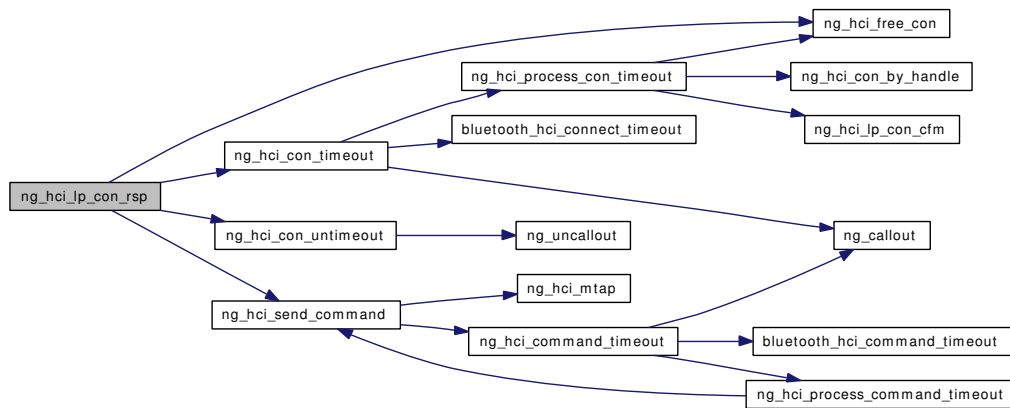
7.40.1.4 int ng_hci_lp_con_rsp (ng_hci_unit_p, item_p, hook_p)

Definition at line 682 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::bdaddr, ng_hci_unit::cmdq, ng_hci_unit::features, ng_hci_unit_con::flags, ng_hci_unit_con::link_type, NG_BT_MBUFQ_ENQUEUE, NG_FREE_ITEM, NG_HCI_ALERT, NG_HCI_CMD_PKT, NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, ng_hci_con_timeout(), ng_hci_con_untimeout(), NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), NG_HCI_LMP_SWITCH, NG_HCI_OCF_ACCEPT_CON, NG_HCI_OCF_REJECT_CON, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OPCODE, NG_HCI_ROLE_MASTER, NG_HCI_ROLE_SLAVE, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_HCI_UNIT_READY, NG_HCI_WARN, NG_NODE_NAME, NGI_MSG, ng_hci_unit::node, packed, ng_hci_unit::role_switch, ng_hci_unit_con::state, and ng_hci_unit::state.

Referenced by ng_hci_upper_rcvmsg().

Here is the call graph for this function:



7.40.1.5 `int ng_hci_lp_discon_ind (ng_hci_unit_con_p, int)`

Definition at line 876 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::con_handle`, `ng_mesg::data`, `ng_hci_unit_con::link_type`, `NG_HCI_INFO`, `NG_HCI_LINK_ACL`, `NG_HOOK_IS_VALID`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_DISCON_IND`, `ng_hci_unit::node`, `ng_hci_unit::sco`, and `ng_hci_unit_con::unit`.

Referenced by `discon_compl()`, `ng_hci_unit_clean()`, and `process_hc_baseband_params()`.

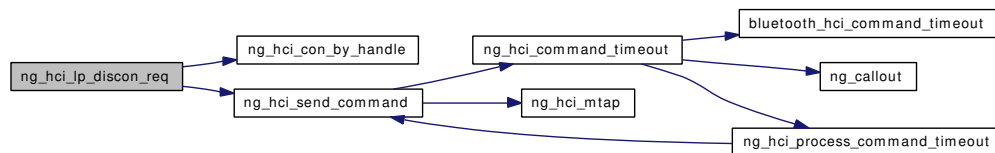
7.40.1.6 `int ng_hci_lp_discon_req (ng_hci_unit_p, item_p, hook_p)`

Definition at line 477 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::cmdq`, `NG_BT_MBUFQ_ENQUEUE`, `NG_FREE_ITEM`, `NG_HCI_ALERT`, `NG_HCI_CMD_PKT`, `ng_hci_con_by_handle()`, `NG_HCI_CON_OPEN`, `NG_HCI_ERR`, `NG_HCI_OCF_DISCON`, `NG_HCI_OGF_LINK_CONTROL`, `NG_HCI_OPCODE`, `ng_hci_send_command()`, `NG_HCI_UNIT_COMMAND_PENDING`, `NG_HCI_UNIT_READY`, `NG_HCI_WARN`, `NG_NODE_NAME`, `NGI_MSG`, `ng_hci_unit::node`, `packed`, `ng_hci_unit_con::state`, and `ng_hci_unit::state`.

Referenced by `ng_hci_upper_rcvmsg()`.

Here is the call graph for this function:



7.40.1.7 `int ng_hci_lp_qos_cfm (ng_hci_unit_con_p, int)`

Definition at line 1047 of file `ng_hci_ulpi.c`.

References `ng_hci_unit::acl`, `ng_hci_unit_con::con_handle`, `ng_mesg::data`, `ng_hci_unit_con::flags`, `NG_HCI_CON_NOTIFY_ACL`, `NG_HCI_CON_NOTIFY_SCO`, `NG_HCI_INFO`, `NG_HOOK_IS_VALID`,

NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_QOS_CFM, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by qos_setup_compl().

7.40.1.8 int ng_hci_lp_qos_ind (ng_hci_unit_con_p)

Definition at line 1102 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit_con::con_handle, ng_mesg::data, NG_HCI_INFO, NG_HOOK_IS_VALID, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_QOS_IND, ng_hci_unit::node, ng_hci_unit::sco, and ng_hci_unit_con::unit.

Referenced by qos_violation().

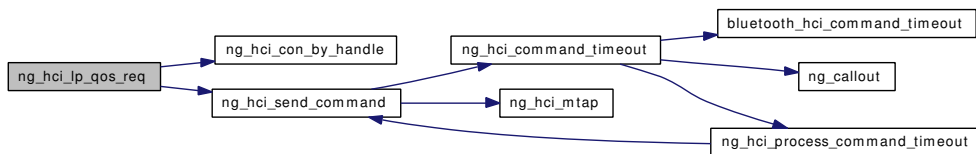
7.40.1.9 int ng_hci_lp_qos_req (ng_hci_unit_p, item_p, hook_p)

Definition at line 934 of file ng_hci_ulpi.c.

References ng_hci_unit::acl, ng_hci_unit::cmdq, ng_hci_unit_con::con_handle, ng_hci_unit_con::flags, ng_hci_unit_con::link_type, NG_BT_MBUFQ_ENQUEUE, NG_FREE_ITEM, NG_HCI_ALERT, NG_HCI_CMD_PKT, ng_hci_con_by_handle(), NG_HCI_CON_NOTIFY_ACL, NG_HCI_CON_NOTIFY_SCO, NG_HCI_CON_OPEN, NG_HCI_ERR, NG_HCI_LINK_ACL, NG_HCI_OCF_QOS_SETUP, NG_HCI_OGF_LINK_POLICY, NG_HCI_OPCODE, ng_hci_send_command(), NG_HCI_UNIT_COMMAND_PENDING, NG_HCI_UNIT_READY, NG_HCI_WARN, NG_NODE_NAME, NGI_MSG, ng_hci_unit::node, packed, ng_hci_unit_con::state, and ng_hci_unit::state.

Referenced by ng_hci_upper_rcvmsg().

Here is the call graph for this function:



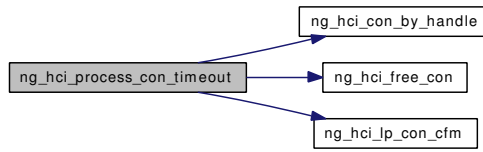
7.40.1.10 void ng_hci_process_con_timeout (node_p, hook_p, void *, int)

Definition at line 1152 of file ng_hci_ulpi.c.

References ng_hci_unit_con::flags, NG_HCI_ALERT, ng_hci_con_by_handle(), NG_HCI_CON_TIMEOUT_PENDING, NG_HCI_CON_W4_CONN_COMPLETE, NG_HCI_CON_W4_LP_CON_RSP, ng_hci_free_con(), ng_hci_lp_con_cfm(), NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, and ng_hci_unit_con::state.

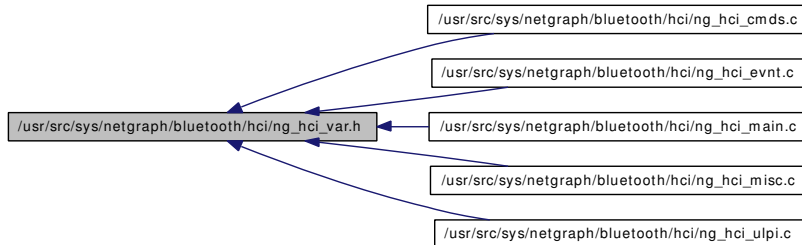
Referenced by ng_hci_con_timeout().

Here is the call graph for this function:



7.41 /usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_hci_unit_buff](#)
- struct [ng_hci_unit](#)
- struct [ng_hci_unit_con](#)
- struct [ng_hci_neighbor](#)

Defines

- #define [M_NETGRAPH_HCI](#) M_NETGRAPH
- #define [NG_HCI_ALERT](#) if (unit → debug >= NG_HCI_ALERT_LEVEL) printf
- #define [NG_HCI_ERR](#) if (unit → debug >= NG_HCI_ERR_LEVEL) printf
- #define [NG_HCI_WARN](#) if (unit → debug >= NG_HCI_WARN_LEVEL) printf
- #define [NG_HCI_INFO](#) if (unit → debug >= NG_HCI_INFO_LEVEL) printf
- #define [NG_HCI_M_PULLUP](#)(m, s)
- #define [NG_HCI_BUFF_CMD_SET](#)(b, v) (b).cmd_free = (v)
- #define [NG_HCI_BUFF_CMD_GET](#)(b, v) (v) = (b).cmd_free
- #define [NG_HCI_BUFF_CMD_USE](#)(b, v) (b).cmd_free -= (v)
- #define [NG_HCI_BUFF_ACL_USE](#)(b, v) (b).acl_free -= (v)
- #define [NG_HCI_BUFF_ACL_FREE](#)(b, v)
- #define [NG_HCI_BUFF_ACL_AVAIL](#)(b, v) (v) = (b).acl_free
- #define [NG_HCI_BUFF_ACL_TOTAL](#)(b, v) (v) = (b).acl_pkts
- #define [NG_HCI_BUFF_ACL_SIZE](#)(b, v) (v) = (b).acl_size
- #define [NG_HCI_BUFF_ACL_SET](#)(b, n, s, f)
- #define [NG_HCI_BUFF_SCO_USE](#)(b, v) (b).sco_free -= (v)
- #define [NG_HCI_BUFF_SCO_FREE](#)(b, v)
- #define [NG_HCI_BUFF_SCO_AVAIL](#)(b, v) (v) = (b).sco_free
- #define [NG_HCI_BUFF_SCO_TOTAL](#)(b, v) (v) = (b).sco_pkts
- #define [NG_HCI_BUFF_SCO_SIZE](#)(b, v) (v) = (b).sco_size
- #define [NG_HCI_BUFF_SCO_SET](#)(b, n, s, f)
- #define [NG_HCI_STAT_CMD_SENT](#)(s) (s).cmd_sent ++
- #define [NG_HCI_STAT_EVT_RECV](#)(s) (s).evnt_recv ++
- #define [NG_HCI_STAT_ACL_SENT](#)(s, n) (s).acl_sent += (n)
- #define [NG_HCI_STAT_ACL_RECV](#)(s) (s).acl_recv ++

- #define `NG_HCI_STAT_SCO_SENT`(s, n) (s).sco_sent += (n)
- #define `NG_HCI_STAT_SCO_RECV`(s) (s).sco_recv ++
- #define `NG_HCI_STAT_BYTES_SENT`(s, b) (s).bytes_sent += (b)
- #define `NG_HCI_STAT_BYTES_RECV`(s, b) (s).bytes_recv += (b)
- #define `NG_HCI_STAT_RESET`(s) bzero(&(s), sizeof((s)))
- #define `NG_HCI_CMD_QUEUE_LEN` 12
- #define `NG_HCI_CON_TIMEOUT_PENDING` (1 << 0)
- #define `NG_HCI_CON_NOTIFY_ACL` (1 << 1)
- #define `NG_HCI_CON_NOTIFY_SCO` (1 << 2)

Typedefs

- typedef `ng_hci_unit_buff` `ng_hci_unit_buff_t`
- typedef `ng_hci_unit` `ng_hci_unit_t`
- typedef `ng_hci_unit_t *` `ng_hci_unit_p`
- typedef `ng_hci_unit_con` `ng_hci_unit_con_t`
- typedef `ng_hci_unit_con_t *` `ng_hci_unit_con_p`
- typedef `ng_hci_neighbor` `ng_hci_neighbor_t`
- typedef `ng_hci_neighbor_t *` `ng_hci_neighbor_p`

7.41.1 Define Documentation

7.41.1.1 #define M_NETGRAPH_HCI M_NETGRAPH

Definition at line 41 of file `ng_hci_var.h`.

7.41.1.2 #define NG_HCI_ALERT if (unit → debug >= NG_HCI_ALERT_LEVEL) printf

Definition at line 45 of file `ng_hci_var.h`.

Referenced by `complete_command()`, `data_buffer_overflow()`, `discon_compl()`, `encryption_change()`, `hardware_error()`, `mode_change()`, `ng_hci_acl_rcvdata()`, `ng_hci_drv_rcvdata()`, `ng_hci_lp_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_process_command_timeout()`, `ng_hci_process_con_timeout()`, `ng_hci_raw_rcvdata()`, `ng_hci_sco_rcvdata()`, `num_compl_pkts()`, `process_link_policy_params()`, `qos_setup_compl()`, `qos_violation()`, `read_clock_offset_compl()`, `read_remote_features_compl()`, and `role_change()`.

7.41.1.3 #define NG_HCI_BUFFER_ACL_AVAIL(b, v) (v) = (b).acl_free

Definition at line 92 of file `ng_hci_var.h`.

Referenced by `ng_hci_default_rcvmsg()`, and `ng_hci_send_data()`.

7.41.1.4 #define NG_HCI_BUFFER_ACL_FREE(b, v)

Value:

```
do {
    (b).acl_free += (v);
    if ((b).acl_free > (b).acl_pkts)
        (b).acl_free = (b).acl_pkts;
} while (0)
```

Definition at line 86 of file ng_hci_var.h.

Referenced by ng_hci_free_con(), ng_hci_unit_clean(), num_compl_pkts(), and process_hc_baseband_params().

7.41.1.5 #define NG_HCI_BUFF_ACL_SET(b, n, s, f)

Value:

```
do {
    (b).acl_free = (f);
    (b).acl_size = (s);
    (b).acl_pkts = (n);
} while (0)
```

Definition at line 95 of file ng_hci_var.h.

Referenced by ng_hci_constructor(), and process_info_params().

7.41.1.6 #define NG_HCI_BUFF_ACL_SIZE(b, v) (v) = (b).acl_size

Definition at line 94 of file ng_hci_var.h.

Referenced by ng_hci_acl_rcvdata(), ng_hci_default_rcvmsg(), and ng_hci_node_is_up().

7.41.1.7 #define NG_HCI_BUFF_ACL_TOTAL(b, v) (v) = (b).acl_pkts

Definition at line 93 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_new_con(), ng_hci_node_is_up(), ng_hci_unit_clean(), and process_hc_baseband_params().

7.41.1.8 #define NG_HCI_BUFF_ACL_USE(b, v) (b).acl_free -= (v)

Definition at line 85 of file ng_hci_var.h.

Referenced by ng_hci_send_data().

7.41.1.9 #define NG_HCI_BUFF_CMD_GET(b, v) (v) = (b).cmd_free

Definition at line 82 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_send_command().

7.41.1.10 #define NG_HCI_BUFF_CMD_SET(b, v) (b).cmd_free = (v)

Definition at line 81 of file ng_hci_var.h.

Referenced by ng_hci_constructor(), ng_hci_process_command_complete(), ng_hci_process_command_status(), ng_hci_process_command_timeout(), and ng_hci_unit_clean().

7.41.1.11 #define NG_HCI_BUFF_CMD_USE(b, v) (b).cmd_free -= (v)

Definition at line 83 of file ng_hci_var.h.

Referenced by ng_hci_send_command().

7.41.1.12 #define NG_HCI_BUFF_SCO_AVAIL(b, v) (v) = (b).sco_free

Definition at line 109 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_send_data().

7.41.1.13 #define NG_HCI_BUFF_SCO_FREE(b, v)**Value:**

```
do {
    (b).sco_free += (v);
    if ((b).sco_free > (b).sco_pkts)
        (b).sco_free = (b).sco_pkts;
} while (0)
```

Definition at line 103 of file ng_hci_var.h.

Referenced by ng_hci_free_con(), ng_hci_unit_clean(), num_compl_pkts(), and process_hc_baseband_params().

7.41.1.14 #define NG_HCI_BUFF_SCO_SET(b, n, s, f)**Value:**

```
do {
    (b).sco_free = (f);
    (b).sco_size = (s);
    (b).sco_pkts = (n);
} while (0)
```

Definition at line 112 of file ng_hci_var.h.

Referenced by ng_hci_constructor(), and process_info_params().

7.41.1.15 #define NG_HCI_BUFF_SCO_SIZE(b, v) (v) = (b).sco_size

Definition at line 111 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_node_is_up(), and ng_hci_sco_rcvdata().

7.41.1.16 #define NG_HCI_BUFF_SCO_TOTAL(b, v) (v) = (b).sco_pkts

Definition at line 110 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_new_con(), ng_hci_node_is_up(), ng_hci_unit_clean(), and process_hc_baseband_params().

7.41.1.17 #define NG_HCI_BUFF_SCO_USE(b, v) (b).sco_free -= (v)

Definition at line 102 of file ng_hci_var.h.

Referenced by ng_hci_send_data().

7.41.1.18 #define NG_HCI_CMD_QUEUE_LEN 12

Definition at line 155 of file ng_hci_var.h.

Referenced by ng_hci_constructor().

7.41.1.19 #define NG_HCI_CON_NOTIFY_ACL (1 << 1)

Definition at line 177 of file ng_hci_var.h.

Referenced by ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_rsp(), ng_hci_lp_qos_cfm(), and ng_hci_lp_qos_req().

7.41.1.20 #define NG_HCI_CON_NOTIFY_SCO (1 << 2)

Definition at line 178 of file ng_hci_var.h.

Referenced by ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_rsp(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_req(), and ng_hci_lp_sco_con_req().

7.41.1.21 #define NG_HCI_CON_TIMEOUT_PENDING (1 << 0)

Definition at line 176 of file ng_hci_var.h.

Referenced by discon_compl(), ng_hci_con_timeout(), ng_hci_con_untimeout(), ng_hci_process_con_timeout(), ng_hci_unit_clean(), and process_hc_baseband_params().

7.41.1.22 #define NG_HCI_ERR if (unit → debug >= NG_HCI_ERR_LEVEL) printf

Definition at line 46 of file ng_hci_var.h.

Referenced by encryption_change(), mode_change(), ng_hci_acl_rcvdata(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_process_command_complete(), ng_hci_process_command_timeout(), ng_hci_sco_rcvdata(), ng_hci_send_command(), read_clock_offset_compl(), read_remote_features_compl(), role_change(), and send_data_packets().

7.41.1.23 #define NG_HCI_INFO if (unit → debug >= NG_HCI_INFO_LEVEL) printf

Definition at line 48 of file ng_hci_var.h.

Referenced by ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_discon_ind(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_ind(), ng_hci_lp_sco_con_req(), ng_hci_mtap(), ng_hci_node_is_up(), ng_hci_process_event(), ng_hci_send_data(), and send_data_packets().

7.41.1.24 #define NG_HCI_M_PULLUP(m, s)

Value:

```

do {
    if ((m)->m_len < (s))
        (m) = m_pullup((m), (s));
    if ((m) == NULL)
        NG_HCI_ALERT("%s: %s - m_pullup(%zd) failed\n", \
            __func__, NG_NODE_NAME(unit->node), (s)); \
} while (0)

```

Definition at line 51 of file ng_hci_var.h.

Referenced by con_compl(), con_req(), discon_compl(), encryption_change(), inquiry_result(), mode_change(), ng_hci_acl_rcvdata(), ng_hci_process_command_complete(), ng_hci_process_command_status(), ng_hci_process_event(), ng_hci_raw_rcvdata(), ng_hci_sco_rcvdata(), num_compl_pkts(), page_scan_mode_change(), page_scan_rep_mode_change(), process_info_params(), process_link_policy_params(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), read_remote_features_compl(), and role_change().

7.41.1.25 #define NG_HCI_STAT_ACL_RECV(s) (s).acl_recv ++

Definition at line 144 of file ng_hci_var.h.

Referenced by ng_hci_drv_rcvdata().

7.41.1.26 #define NG_HCI_STAT_ACL_SENT(s, n) (s).acl_sent += (n)

Definition at line 143 of file ng_hci_var.h.

Referenced by ng_hci_send_data().

7.41.1.27 #define NG_HCI_STAT_BYTES_RECV(s, b) (s).bytes_recv += (b)

Definition at line 148 of file ng_hci_var.h.

Referenced by ng_hci_drv_rcvdata().

7.41.1.28 #define NG_HCI_STAT_BYTES_SENT(s, b) (s).bytes_sent += (b)

Definition at line 147 of file ng_hci_var.h.

Referenced by ng_hci_send_command(), and send_data_packets().

7.41.1.29 #define NG_HCI_STAT_CMD_SENT(s) (s).cmd_sent ++

Definition at line 141 of file ng_hci_var.h.

Referenced by ng_hci_send_command().

7.41.1.30 #define NG_HCI_STAT_EVT_RECV(s) (s).evnt_recv ++

Definition at line 142 of file ng_hci_var.h.

Referenced by ng_hci_drv_rcvdata().

7.41.1.31 #define NG_HCI_STAT_RESET(s) bzero(&(s), sizeof((s)))

Definition at line 149 of file ng_hci_var.h.

Referenced by ng_hci_default_rcvmsg().

7.41.1.32 #define NG_HCI_STAT_SCO_RECV(s) (s).sco_recv ++

Definition at line 146 of file ng_hci_var.h.

Referenced by ng_hci_drv_rcvdata().

7.41.1.33 #define NG_HCI_STAT_SCO_SENT(s, n) (s).sco_sent += (n)

Definition at line 145 of file ng_hci_var.h.

Referenced by ng_hci_send_data().

7.41.1.34 #define NG_HCI_WARN if (unit → debug >= NG_HCI_WARN_LEVEL) printf

Definition at line 47 of file ng_hci_var.h.

Referenced by ng_hci_drv_rcvdata(), ng_hci_lp_con_ind(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_send_command(), and num_compl_pkts().

7.41.2 Typedef Documentation**7.41.2.1 typedef ng_hci_neighbor_t* ng_hci_neighbor_p**

Definition at line 215 of file ng_hci_var.h.

7.41.2.2 typedef struct ng_hci_neighbor ng_hci_neighbor_t**7.41.2.3 typedef struct ng_hci_unit_buff ng_hci_unit_buff_t****7.41.2.4 typedef ng_hci_unit_con_t* ng_hci_unit_con_p**

Definition at line 195 of file ng_hci_var.h.

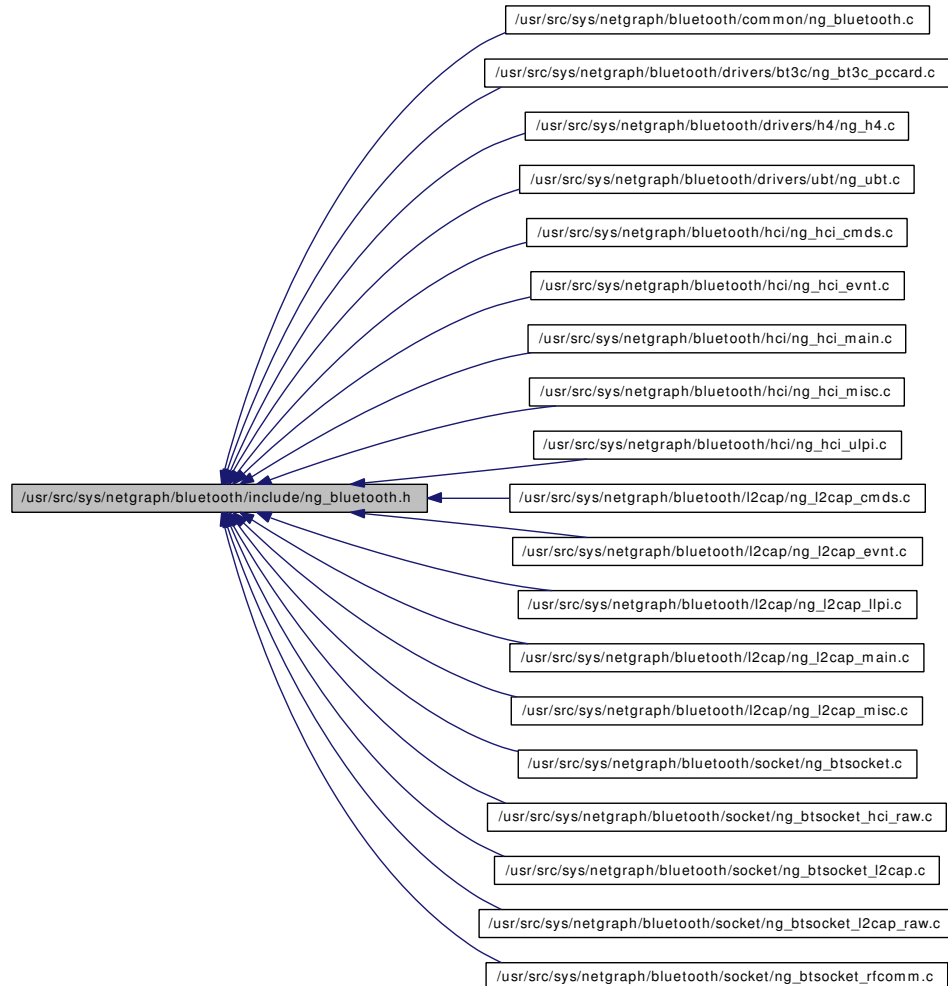
7.41.2.5 typedef struct ng_hci_unit_con ng_hci_unit_con_t**7.41.2.6 typedef ng_hci_unit_t* ng_hci_unit_p**

Definition at line 165 of file ng_hci_var.h.

7.41.2.7 typedef struct ng_hci_unit ng_hci_unit_t

7.42 /usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_bt_mbufq](#)
- struct [ng_bt_itemq](#)

Defines

- #define [NG_BLUETOOTH_VERSION](#) 1
- #define [NG_BT_MBUFQ_INIT](#)(q, _maxlen)
- #define [NG_BT_MBUFQ_DESTROY](#)(q)
- #define [NG_BT_MBUFQ_FIRST](#)(q) (q) → head
- #define [NG_BT_MBUFQ_LEN](#)(q) (q) → len
- #define [NG_BT_MBUFQ_FULL](#)(q) ((q) → len >= (q) → maxlen)

- #define `NG_BT_MBUFQ_DROP(q)` (q) → drops ++
- #define `NG_BT_MBUFQ_ENQUEUE(q, i)`
- #define `NG_BT_MBUFQ_DEQUEUE(q, i)`
- #define `NG_BT_MBUFQ_PREPEND(q, i)`
- #define `NG_BT_MBUFQ_DRAIN(q)`
- #define `NG_BT_ITEMQ_INIT(q, _maxlen)` `NG_BT_MBUFQ_INIT((q), (_maxlen))`
- #define `NG_BT_ITEMQ_DESTROY(q)`
- #define `NG_BT_ITEMQ_FIRST(q)` `NG_BT_MBUFQ_FIRST((q))`
- #define `NG_BT_ITEMQ_LEN(q)` `NG_BT_MBUFQ_LEN((q))`
- #define `NG_BT_ITEMQ_FULL(q)` `NG_BT_MBUFQ_FULL((q))`
- #define `NG_BT_ITEMQ_DROP(q)` `NG_BT_MBUFQ_DROP((q))`
- #define `NG_BT_ITEMQ_ENQUEUE(q, i)`
- #define `NG_BT_ITEMQ_DEQUEUE(q, i)`
- #define `NG_BT_ITEMQ_PREPEND(q, i)`
- #define `NG_BT_ITEMQ_DRAIN(q)`

Typedefs

- typedef `ng_bt_mbufq` `ng_bt_mbufq_t`
- typedef `ng_bt_mbufq *` `ng_bt_mbufq_p`
- typedef `ng_bt_itemq` `ng_bt_itemq_t`
- typedef `ng_bt_itemq *` `ng_bt_itemq_p`

Functions

- `u_int32_t` `bluetooth_hci_command_timeout` (void)
- `u_int32_t` `bluetooth_hci_connect_timeout` (void)
- `u_int32_t` `bluetooth_hci_max_neighbor_age` (void)
- `u_int32_t` `bluetooth_l2cap_rtx_timeout` (void)
- `u_int32_t` `bluetooth_l2cap_ertx_timeout` (void)

7.42.1 Define Documentation

7.42.1.1 #define NG_BLUETOOTH_VERSION 1

Definition at line 41 of file `ng_bluetooth.h`.

7.42.1.2 #define NG_BT_ITEMQ_DEQUEUE(q, i)

Value:

```
do {
    (i) = (q)->head;
    if ((i) != NULL) {
        (q)->head = (q)->head->el_next;
        if ((q)->head == NULL)
            (q)->tail = NULL;

        (q)->len --;
        (i)->el_next = NULL;
    }
} while (0)
```

Definition at line 187 of file ng_bluetooth.h.

Referenced by ng_btsocket_hci_raw_input(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_raw_input(), and send_data_packets().

7.42.1.3 #define NG_BT_ITEMQ_DESTROY(q)

Value:

```
do {
    NG_BT_ITEMQ_DRAIN((q));
} while (0)
```

Definition at line 161 of file ng_bluetooth.h.

Referenced by ng_hci_free_con().

7.42.1.4 #define NG_BT_ITEMQ_DRAIN(q)

Value:

```
do {
    struct ng_item *i = NULL;
    for (;;) {
        NG_BT_ITEMQ_DEQUEUE((q), i);
        if (i == NULL)
            break;
        NG_FREE_ITEM(i);
    }
} while (0)
```

Definition at line 210 of file ng_bluetooth.h.

7.42.1.5 #define NG_BT_ITEMQ_DROP(q) NG_BT_MBUFQ_DROP((q))

Definition at line 172 of file ng_bluetooth.h.

Referenced by ng_btsocket_hci_raw_node_rcvdata(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_node_rcvdata(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_hci_acl_rcvdata(), and ng_hci_sco_rcvdata().

7.42.1.6 #define NG_BT_ITEMQ_ENQUEUE(q, i)

Value:

```
do {
    (i)->el_next = NULL;
    if ((q)->tail == NULL)
        (q)->head = (i);
    else
        (q)->tail->el_next = (i);
    (q)->tail = (i);
    (q)->len ++;
} while (0)
```

Definition at line 174 of file ng_bluetooth.h.

Referenced by ng_btsocket_hci_raw_node_rcvdata(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_node_rcvdata(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_hci_acl_rcvdata(), and ng_hci_sco_rcvdata().

7.42.1.7 #define NG_BT_ITEMQ_FIRST(q) NG_BT_MBUFQ_FIRST((q))

Definition at line 166 of file ng_bluetooth.h.

7.42.1.8 #define NG_BT_ITEMQ_FULL(q) NG_BT_MBUFQ_FULL((q))

Definition at line 170 of file ng_bluetooth.h.

Referenced by ng_btsocket_hci_raw_node_rcvdata(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_node_rcvdata(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_hci_acl_rcvdata(), and ng_hci_sco_rcvdata().

7.42.1.9 #define NG_BT_ITEMQ_INIT(q, _maxlen) NG_BT_MBUFQ_INIT((q), (_maxlen))

Definition at line 159 of file ng_bluetooth.h.

Referenced by ng_btsocket_hci_raw_init(), ng_btsocket_l2cap_init(), ng_btsocket_l2cap_raw_init(), and ng_hci_new_con().

7.42.1.10 #define NG_BT_ITEMQ_LEN(q) NG_BT_MBUFQ_LEN((q))

Definition at line 168 of file ng_bluetooth.h.

Referenced by ng_hci_acl_rcvdata(), ng_hci_default_rcvmsg(), ng_hci_sco_rcvdata(), and send_data_packets().

7.42.1.11 #define NG_BT_ITEMQ_PREPEND(q, i)

Value:

```
do {
    (i)->el_next = (q)->head;          \
    if ((q)->tail == NULL)             \
        (q)->tail = (i);              \
                                        \
    (q)->head = (i);                   \
    (q)->len ++;                       \
} while (0)
```

Definition at line 200 of file ng_bluetooth.h.

7.42.1.12 #define NG_BT_MBUFQ_DEQUEUE(q, i)

Value:

```
do {
    (i) = (q)->head;                   \
                                        \

```

```

        if ((i) != NULL) {
            (q)->head = (q)->head->m_nextpkt; \
            if ((q)->head == NULL) \
                (q)->tail = NULL; \
            (q)->len --; \
            (i)->m_nextpkt = NULL; \
        } \
    } while (0)

```

Definition at line 107 of file ng_bluetooth.h.

Referenced by complete_command(), ng_btsocket_rfcomm_session_send(), ng_h4_start2(), ng_hci_process_command_timeout(), ubt_bulk_out_start(), ubt_isoc_out_start(), and ubt_request_start().

7.42.1.13 #define NG_BT_MBUFQ_DESTROY(q)

Value:

```

do {
    NG_BT_MBUFQ_DRAIN((q)); \
} while (0)

```

Definition at line 81 of file ng_bluetooth.h.

Referenced by ng_h4_close(), and ng_hci_shutdown().

7.42.1.14 #define NG_BT_MBUFQ_DRAIN(q)

Value:

```

do {
    struct mbuf *m = NULL; \
    for (;;) { \
        NG_BT_MBUFQ_DEQUEUE((q), m); \
        if (m == NULL) \
            break; \
        NG_FREE_M(m); \
    } \
} while (0)

```

Definition at line 130 of file ng_bluetooth.h.

Referenced by ng_btsocket_rfcomm_sessions_task(), ng_h4_disconnect(), ng_h4_rcvmsg(), ng_hci_send_command(), ng_hci_unit_clean(), ubt_reset(), and USB_DETACH().

7.42.1.15 #define NG_BT_MBUFQ_DROP(q) (q) → drops ++

Definition at line 92 of file ng_bluetooth.h.

Referenced by ng_h4_rcvdata(), ng_hci_raw_rcvdata(), ubt_bulk_out_start(), ubt_isoc_out_start(), and ubt_request_start().

7.42.1.16 #define NG_BT_MBUFQ_ENQUEUE(q, i)**Value:**

```

do {
    (i)->m_nextpkt = NULL;
    if ((q)->tail == NULL)
        (q)->head = (i);
    else
        (q)->tail->m_nextpkt = (i);
    (q)->tail = (i);
    (q)->len ++;
} while (0)

```

Definition at line 94 of file ng_bluetooth.h.

Referenced by con_compl(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_uih(), ng_h4_rcvdata(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_raw_rcvdata(), and ng_ubt_rcvdata().

7.42.1.17 #define NG_BT_MBUFQ_FIRST(q) (q) → head

Definition at line 86 of file ng_bluetooth.h.

Referenced by complete_command(), and ng_hci_send_command().

7.42.1.18 #define NG_BT_MBUFQ_FULL(q) ((q) → len >= (q) → maxlen)

Definition at line 90 of file ng_bluetooth.h.

Referenced by ng_h4_rcvdata(), ng_hci_raw_rcvdata(), and ng_ubt_rcvdata().

7.42.1.19 #define NG_BT_MBUFQ_INIT(q, _maxlen)**Value:**

```

do {
    (q)->head = NULL;
    (q)->tail = NULL;
    (q)->len = 0;
    (q)->maxlen = (_maxlen);
    (q)->drops = 0;
} while (0)

```

Definition at line 72 of file ng_bluetooth.h.

Referenced by ng_btsocket_rfcomm_session_create(), ng_h4_open(), ng_hci_constructor(), and USB_ATTACH().

7.42.1.20 #define NG_BT_MBUFQ_LEN(q) (q) → len

Definition at line 88 of file ng_bluetooth.h.

Referenced by ng_h4_rcvmsg(), ng_h4_start2(), ng_hci_default_rcvmsg(), ng_hci_raw_rcvdata(), ng_ubt_rcvmsg(), ubt_bulk_out_complete2(), ubt_isoc_out_complete2(), and ubt_request_complete2().

7.42.1.21 #define NG_BT_MBUFQ_PREPEND(q, i)**Value:**

```
do {
    (i)->m_nextpkt = (q)->head;
    if ((q)->tail == NULL)
        (q)->tail = (i);
    (q)->head = (i);
    (q)->len ++;
} while (0)
```

Definition at line 120 of file ng_bluetooth.h.

Referenced by ng_h4_start2().

7.42.2 Typedef Documentation**7.42.2.1 typedef struct [ng_bt_itemq*](#) [ng_bt_itemq_p](#)**

Definition at line 157 of file ng_bluetooth.h.

7.42.2.2 typedef struct [ng_bt_itemq](#) [ng_bt_itemq_t](#)

Definition at line 156 of file ng_bluetooth.h.

7.42.2.3 typedef struct [ng_bt_mbufq*](#) [ng_bt_mbufq_p](#)

Definition at line 70 of file ng_bluetooth.h.

7.42.2.4 typedef struct [ng_bt_mbufq](#) [ng_bt_mbufq_t](#)

Definition at line 69 of file ng_bluetooth.h.

7.42.3 Function Documentation**7.42.3.1 [u_int32_t](#) [bluetooth_hci_command_timeout](#) (void)**

Definition at line 181 of file ng_bluetooth.c.

References [bluetooth_hci_command_timeout_value](#).

Referenced by [ng_hci_command_timeout\(\)](#).

7.42.3.2 [u_int32_t](#) [bluetooth_hci_connect_timeout](#) (void)

Definition at line 187 of file ng_bluetooth.c.

References [bluetooth_hci_connect_timeout_value](#).

Referenced by [ng_hci_con_timeout\(\)](#), and [ng_l2cap_lp_timeout\(\)](#).

7.42.3.3 u_int32_t bluetooth_hci_max_neighbor_age (void)

Definition at line 193 of file ng_bluetooth.c.

References bluetooth_hci_max_neighbor_age_value.

Referenced by ng_hci_neighbor_stale().

7.42.3.4 u_int32_t bluetooth_l2cap_ertx_timeout (void)

Definition at line 205 of file ng_bluetooth.c.

References bluetooth_l2cap_ertx_timeout_value.

Referenced by ng_btsocket_l2cap_timeout(), and ng_l2cap_process_con_rsp().

7.42.3.5 u_int32_t bluetooth_l2cap_rtx_timeout (void)

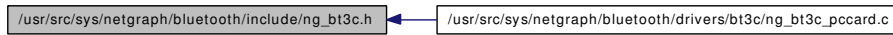
Definition at line 199 of file ng_bluetooth.c.

References bluetooth_l2cap_rtx_timeout_value.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_con_wakeup().

7.43 /usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_bt3c_node_qlen_ep](#)
- struct [ng_bt3c_node_stat_ep](#)
- struct [ng_bt3c_firmware_block_ep](#)

Defines

- #define [NG_BT3C_NODE_TYPE](#) "btccc"
- #define [NG_BT3C_HOOK](#) "hook"
- #define [NGM_BT3C_COOKIE](#) 1014752016
- #define [NG_BT3C_ALERT_LEVEL](#) 1
- #define [NG_BT3C_ERR_LEVEL](#) 2
- #define [NG_BT3C_WARN_LEVEL](#) 3
- #define [NG_BT3C_INFO_LEVEL](#) 4
- #define [NG_BT3C_W4_PKT_IND](#) 1
- #define [NG_BT3C_W4_PKT_HDR](#) 2
- #define [NG_BT3C_W4_PKT_DATA](#) 3
- #define [NGM_BT3C_NODE_GET_STATE](#) 1
- #define [NGM_BT3C_NODE_SET_DEBUG](#) 2
- #define [NGM_BT3C_NODE_GET_DEBUG](#) 3
- #define [NGM_BT3C_NODE_GET_QLEN](#) 4
- #define [NGM_BT3C_NODE_SET_QLEN](#) 5
- #define [NGM_BT3C_NODE_IN_QUEUE](#) 1
- #define [NGM_BT3C_NODE_OUT_QUEUE](#) 2
- #define [NGM_BT3C_NODE_GET_STAT](#) 6
- #define [NGM_BT3C_NODE_RESET_STAT](#) 7
- #define [NGM_BT3C_NODE_DOWNLOAD_FIRMWARE](#) 8

Typedefs

- typedef u_int16_t [ng_bt3c_node_state_ep](#)
- typedef u_int16_t [ng_bt3c_node_debug_ep](#)

7.43.1 Define Documentation

7.43.1.1 #define NG_BT3C_ALERT_LEVEL 1

Definition at line 57 of file [ng_bt3c.h](#).

7.43.1.2 #define NG_BT3C_ERR_LEVEL 2

Definition at line 58 of file ng_bt3c.h.

7.43.1.3 #define NG_BT3C_HOOK "hook"

Definition at line 52 of file ng_bt3c.h.

Referenced by ng_bt3c_newhook(), and ng_bt3c_rcvmsg().

7.43.1.4 #define NG_BT3C_INFO_LEVEL 4

Definition at line 60 of file ng_bt3c.h.

7.43.1.5 #define NG_BT3C_NODE_TYPE "btccc"

Definition at line 51 of file ng_bt3c.h.

Referenced by bt3c_intr(), and bt3c_modevent().

7.43.1.6 #define NG_BT3C_W4_PKT_DATA 3

Definition at line 65 of file ng_bt3c.h.

7.43.1.7 #define NG_BT3C_W4_PKT_HDR 2

Definition at line 64 of file ng_bt3c.h.

7.43.1.8 #define NG_BT3C_W4_PKT_IND 1

Definition at line 63 of file ng_bt3c.h.

Referenced by bt3c_pccard_attach(), and bt3c_receive().

7.43.1.9 #define NG_BT3C_WARN_LEVEL 3

Definition at line 59 of file ng_bt3c.h.

Referenced by bt3c_pccard_attach().

7.43.1.10 #define NGM_BT3C_COOKIE 1014752016

Definition at line 54 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.11 #define NGM_BT3C_NODE_DOWNLOAD_FIRMWARE 8

Definition at line 102 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.12 #define NGM_BT3C_NODE_GET_DEBUG 3

Definition at line 77 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.13 #define NGM_BT3C_NODE_GET_QLEN 4

Definition at line 80 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.14 #define NGM_BT3C_NODE_GET_STAT 6

Definition at line 90 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.15 #define NGM_BT3C_NODE_GET_STATE 1

Definition at line 73 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.16 #define NGM_BT3C_NODE_IN_QUEUE 1

Definition at line 84 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.17 #define NGM_BT3C_NODE_OUT_QUEUE 2

Definition at line 85 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.18 #define NGM_BT3C_NODE_RESET_STAT 7

Definition at line 100 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.19 #define NGM_BT3C_NODE_SET_DEBUG 2

Definition at line 76 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.1.20 #define NGM_BT3C_NODE_SET_QLEN 5

Definition at line 81 of file ng_bt3c.h.

Referenced by ng_bt3c_rcvmsg().

7.43.2 Typedef Documentation

7.43.2.1 `typedef u_int16_t ng_bt3c_node_debug_ep`

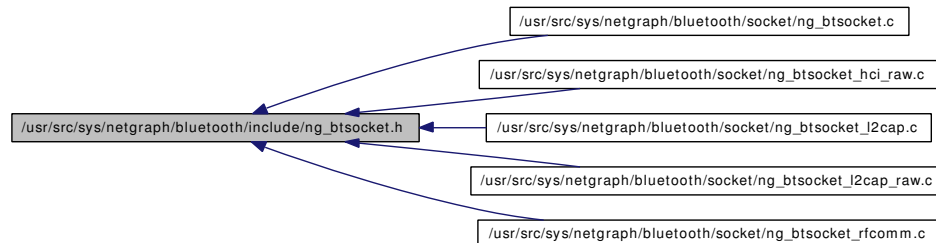
Definition at line 78 of file `ng_bt3c.h`.

7.43.2.2 `typedef u_int16_t ng_bt3c_node_state_ep`

Definition at line 74 of file `ng_bt3c.h`.

7.44 /usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [sockaddr_hci](#)
- struct [ng_btsocket_hci_raw_filter](#)
- struct [ng_btsocket_hci_raw_node_state](#)
- struct [ng_btsocket_hci_raw_node_debug](#)
- struct [ng_btsocket_hci_raw_node_buffer](#)
- struct [ng_btsocket_hci_raw_node_bdaddr](#)
- struct [ng_btsocket_hci_raw_node_features](#)
- struct [ng_btsocket_hci_raw_node_stat](#)
- struct [ng_btsocket_hci_raw_node_neighbor_cache](#)
- struct [ng_btsocket_hci_raw_con_list](#)
- struct [ng_btsocket_hci_raw_node_link_policy_mask](#)
- struct [ng_btsocket_hci_raw_node_packet_mask](#)
- struct [ng_btsocket_hci_raw_node_role_switch](#)
- struct [ng_btsocket_hci_raw_node_list_names](#)
- struct [sockaddr_l2cap](#)
- struct [ng_btsocket_l2cap_raw_ping](#)
- struct [ng_btsocket_l2cap_raw_get_info](#)
- struct [ng_btsocket_l2cap_raw_node_flags](#)
- struct [ng_btsocket_l2cap_raw_node_debug](#)
- struct [ng_btsocket_l2cap_raw_con_list](#)
- struct [ng_btsocket_l2cap_raw_chan_list](#)
- struct [ng_btsocket_l2cap_raw_auto_discon_timo](#)
- struct [sockaddr_rfcomm](#)
- struct [ng_btsocket_rfcomm_fc_info](#)

Defines

- #define [BLUETOOTH_PROTO_HCI](#) 134
- #define [BLUETOOTH_PROTO_L2CAP](#) 135
- #define [BLUETOOTH_PROTO_RFCOMM](#) 136
- #define [SOL_HCI_RAW](#) 0x0802
- #define [SO_HCI_RAW_FILTER](#) 1

- #define SO_HCI_RAW_DIRECTION 2
- #define SCM_HCI_RAW_DIRECTION SO_HCI_RAW_DIRECTION
- #define SIOC_HCI_RAW_NODE_GET_STATE
- #define SIOC_HCI_RAW_NODE_INIT _IO('b', NGM_HCI_NODE_INIT)
- #define SIOC_HCI_RAW_NODE_GET_DEBUG
- #define SIOC_HCI_RAW_NODE_SET_DEBUG
- #define SIOC_HCI_RAW_NODE_GET_BUFFER
- #define SIOC_HCI_RAW_NODE_GET_BDADDR
- #define SIOC_HCI_RAW_NODE_GET_FEATURES
- #define SIOC_HCI_RAW_NODE_GET_STAT
- #define SIOC_HCI_RAW_NODE_RESET_STAT _IO('b', NGM_HCI_NODE_RESET_STAT)
- #define SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_CACHE _IO('b', NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE)
- #define SIOC_HCI_RAW_NODE_GET_NEIGHBOR_CACHE
- #define SIOC_HCI_RAW_NODE_GET_CON_LIST
- #define SIOC_HCI_RAW_NODE_GET_LINK_POLICY_MASK
- #define SIOC_HCI_RAW_NODE_SET_LINK_POLICY_MASK
- #define SIOC_HCI_RAW_NODE_GET_PACKET_MASK
- #define SIOC_HCI_RAW_NODE_SET_PACKET_MASK
- #define SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH
- #define SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH
- #define SIOC_HCI_RAW_NODE_LIST_NAMES
- #define SOL_L2CAP 0x1609
- #define SO_L2CAP_IMTU 1
- #define SO_L2CAP_OMTU 2
- #define SO_L2CAP_IFLOW 3
- #define SO_L2CAP_OFLOW 4
- #define SO_L2CAP_FLUSH 5
- #define SIOC_L2CAP_L2CA_PING
- #define SIOC_L2CAP_L2CA_GET_INFO
- #define SIOC_L2CAP_NODE_GET_FLAGS
- #define SIOC_L2CAP_NODE_GET_DEBUG
- #define SIOC_L2CAP_NODE_SET_DEBUG
- #define SIOC_L2CAP_NODE_GET_CON_LIST
- #define SIOC_L2CAP_NODE_GET_CHAN_LIST
- #define SIOC_L2CAP_NODE_GET_AUTO_DISCON_TIMO
- #define SIOC_L2CAP_NODE_SET_AUTO_DISCON_TIMO
- #define SOL_RFCOMM 0x0816
- #define SO_RFCOMM_MTU 1
- #define SO_RFCOMM_FC_INFO 2
- #define NG_BT_SOCKET_HCI_RAW_NODE_TYPE "btsock_hci_raw"
- #define NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE "btsock_l2c_raw"
- #define NG_BT_SOCKET_L2CAP_NODE_TYPE "btsock_l2c"
- #define NG_BT_SOCKET_ALERT_LEVEL 1
- #define NG_BT_SOCKET_ERR_LEVEL 2
- #define NG_BT_SOCKET_WARN_LEVEL 3
- #define NG_BT_SOCKET_INFO_LEVEL 4

7.44.1 Define Documentation

7.44.1.1 #define BLUETOOTH_PROTO_HCI 134

Definition at line 41 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_attach().

7.44.1.2 #define BLUETOOTH_PROTO_L2CAP 135

Definition at line 42 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_attach(), ng_btsocket_rfcomm_connect(), and ng_btsocket_rfcomm_listen().

7.44.1.3 #define BLUETOOTH_PROTO_RFCOMM 136

Definition at line 43 of file ng_btsocket.h.

Referenced by ng_btsocket_rfcomm_attach().

7.44.1.4 #define NG_BT_SOCKET_ALERT_LEVEL 1

Definition at line 336 of file ng_btsocket.h.

7.44.1.5 #define NG_BT_SOCKET_ERR_LEVEL 2

Definition at line 337 of file ng_btsocket.h.

7.44.1.6 #define NG_BT_SOCKET_HCI_RAW_NODE_TYPE "btsock_hci_raw"

Definition at line 328 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_init(), and ng_btsocket_hci_raw_node_shutdown().

7.44.1.7 #define NG_BT_SOCKET_INFO_LEVEL 4

Definition at line 339 of file ng_btsocket.h.

7.44.1.8 #define NG_BT_SOCKET_L2CAP_NODE_TYPE "btsock_l2c"

Definition at line 330 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_init(), and ng_btsocket_l2cap_node_shutdown().

7.44.1.9 #define NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE "btsock_l2c_raw"

Definition at line 329 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_init(), and ng_btsocket_l2cap_raw_node_shutdown().

7.44.1.10 #define NG_BT_SOCKET_WARN_LEVEL 3

Definition at line 338 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_init(), and ng_btsocket_rfcomm_init().

7.44.1.11 #define SCM_HCI_RAW_DIRECTION SO_HCI_RAW_DIRECTION

Definition at line 60 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_savctl().

**7.44.1.12 #define SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_CACHE_IO('b',
NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE)**

Definition at line 138 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.13 #define SIOC_HCI_RAW_NODE_GET_BDADDR**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_BDADDR, \  
      struct ng_btsocket_hci_raw_node_bdaddr)
```

Definition at line 113 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.14 #define SIOC_HCI_RAW_NODE_GET_BUFFER**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_BUFFER, \  
      struct ng_btsocket_hci_raw_node_buffer)
```

Definition at line 105 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.15 #define SIOC_HCI_RAW_NODE_GET_CON_LIST**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_CON_LIST, \  
      struct ng_btsocket_hci_raw_con_list)
```

Definition at line 155 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.16 #define SIOC_HCI_RAW_NODE_GET_DEBUG**Value:**

```
__IOWR('b', NGM_HCI_NODE_GET_DEBUG, \  
        struct ng_btsocket_hci_raw_node_debug)
```

Definition at line 94 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.17 #define SIOC_HCI_RAW_NODE_GET_FEATURES**Value:**

```
__IOWR('b', NGM_HCI_NODE_GET_FEATURES, \  
        struct ng_btsocket_hci_raw_node_features)
```

Definition at line 121 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.18 #define SIOC_HCI_RAW_NODE_GET_LINK_POLICY_MASK**Value:**

```
__IOWR('b', NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK, \  
        struct ng_btsocket_hci_raw_node_link_policy_mask)
```

Definition at line 163 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.19 #define SIOC_HCI_RAW_NODE_GET_NEIGHBOR_CACHE**Value:**

```
__IOWR('b', NGM_HCI_NODE_GET_NEIGHBOR_CACHE, \  
        struct ng_btsocket_hci_raw_node_neighbor_cache)
```

Definition at line 146 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.20 #define SIOC_HCI_RAW_NODE_GET_PACKET_MASK**Value:**

```
__IOWR('b', NGM_HCI_NODE_GET_PACKET_MASK, \  
        struct ng_btsocket_hci_raw_node_packet_mask)
```

Definition at line 174 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.21 #define SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_ROLE_SWITCH, \  
      struct ng_btsocket_hci_raw_node_role_switch)
```

Definition at line 185 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.22 #define SIOC_HCI_RAW_NODE_GET_STAT**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_STAT, \  
      struct ng_btsocket_hci_raw_node_stat)
```

Definition at line 129 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.23 #define SIOC_HCI_RAW_NODE_GET_STATE**Value:**

```
_IOWR('b', NGM_HCI_NODE_GET_STATE, \  
      struct ng_btsocket_hci_raw_node_state)
```

Definition at line 82 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.24 #define SIOC_HCI_RAW_NODE_INIT_IO('b', NGM_HCI_NODE_INIT)

Definition at line 87 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.25 #define SIOC_HCI_RAW_NODE_LIST_NAMES**Value:**

```
_IOWR('b', NGM_HCI_NODE_LIST_NAMES, \  
      struct ng_btsocket_hci_raw_node_list_names)
```

Definition at line 197 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.26 #define SIOC_HCI_RAW_NODE_RESET_STAT_IO('b', NGM_HCI_NODE_RESET_STAT)

Definition at line 134 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.27 #define SIOC_HCI_RAW_NODE_SET_DEBUG**Value:**

```
__IOWR('b', NGM_HCI_NODE_SET_DEBUG, \  
        struct ng_btsocket_hci_raw_node_debug)
```

Definition at line 97 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.28 #define SIOC_HCI_RAW_NODE_SET_LINK_POLICY_MASK**Value:**

```
__IOWR('b', NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK, \  
        struct ng_btsocket_hci_raw_node_link_policy_mask)
```

Definition at line 166 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.29 #define SIOC_HCI_RAW_NODE_SET_PACKET_MASK**Value:**

```
__IOWR('b', NGM_HCI_NODE_SET_PACKET_MASK, \  
        struct ng_btsocket_hci_raw_node_packet_mask)
```

Definition at line 177 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.30 #define SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH**Value:**

```
__IOWR('b', NGM_HCI_NODE_SET_ROLE_SWITCH, \  
        struct ng_btsocket_hci_raw_node_role_switch)
```

Definition at line 188 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_control().

7.44.1.31 #define SIOC_L2CAP_L2CA_GET_INFO**Value:**

```
__IOWR('b', NGM_L2CAP_L2CA_GET_INFO, \  
        struct ng_btsocket_l2cap_raw_get_info)
```

Definition at line 243 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.32 #define SIOC_L2CAP_L2CA_PING**Value:**

```
__IOWR('b', NGM_L2CAP_L2CA_PING, \  
        struct ng_btsocket_l2cap_raw_ping)
```

Definition at line 232 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.33 #define SIOC_L2CAP_NODE_GET_AUTO_DISCON_TIMO**Value:**

```
__IOWR('b', NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO, \  
        struct ng_btsocket_l2cap_raw_auto_discon_timo)
```

Definition at line 289 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.34 #define SIOC_L2CAP_NODE_GET_CHAN_LIST**Value:**

```
__IOWR('b', NGM_L2CAP_NODE_GET_CHAN_LIST, \  
        struct ng_btsocket_l2cap_raw_chan_list)
```

Definition at line 280 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.35 #define SIOC_L2CAP_NODE_GET_CON_LIST**Value:**

```
__IOWR('b', NGM_L2CAP_NODE_GET_CON_LIST, \  
        struct ng_btsocket_l2cap_raw_con_list)
```

Definition at line 271 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.36 #define SIOC_L2CAP_NODE_GET_DEBUG**Value:**

```
__IOWR('b', NGM_L2CAP_NODE_GET_DEBUG, \  
        struct ng_btsocket_l2cap_raw_node_debug)
```

Definition at line 259 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.37 #define SIOC_L2CAP_NODE_GET_FLAGS**Value:**

```
_IOWR('b', NGM_L2CAP_NODE_GET_FLAGS, \
      struct ng_btsocket_l2cap_raw_node_flags)
```

Definition at line 251 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.38 #define SIOC_L2CAP_NODE_SET_AUTO_DISCON_TIMO**Value:**

```
_IOWR('b', NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO, \
      struct ng_btsocket_l2cap_raw_auto_discon_timo)
```

Definition at line 292 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.39 #define SIOC_L2CAP_NODE_SET_DEBUG**Value:**

```
_IOWR('b', NGM_L2CAP_NODE_SET_DEBUG, \
      struct ng_btsocket_l2cap_raw_node_debug)
```

Definition at line 262 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_raw_control().

7.44.1.40 #define SO_HCI_RAW_DIRECTION 2

Definition at line 59 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_ctloutput().

7.44.1.41 #define SO_HCI_RAW_FILTER 1

Definition at line 58 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_ctloutput().

7.44.1.42 #define SO_L2CAP_FLUSH 5

Definition at line 220 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput().

7.44.1.43 #define SO_L2CAP_IFLOW 3

Definition at line 218 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput().

7.44.1.44 #define SO_L2CAP_IMTU 1

Definition at line 216 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput(), and ng_btsocket_rfcomm_session_create().

7.44.1.45 #define SO_L2CAP_OFLOW 4

Definition at line 219 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput().

7.44.1.46 #define SO_L2CAP_OMTU 2

Definition at line 217 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput().

7.44.1.47 #define SO_RFCOMM_FC_INFO 2

Definition at line 322 of file ng_btsocket.h.

Referenced by ng_btsocket_rfcomm_ctloutput().

7.44.1.48 #define SO_RFCOMM_MTU 1

Definition at line 321 of file ng_btsocket.h.

Referenced by ng_btsocket_rfcomm_ctloutput().

7.44.1.49 #define SOL_HCI_RAW 0x0802

Definition at line 56 of file ng_btsocket.h.

Referenced by ng_btsocket_hci_raw_ctloutput(), and ng_btsocket_hci_raw_savctl().

7.44.1.50 #define SOL_L2CAP 0x1609

Definition at line 214 of file ng_btsocket.h.

Referenced by ng_btsocket_l2cap_ctloutput(), and ng_btsocket_rfcomm_session_create().

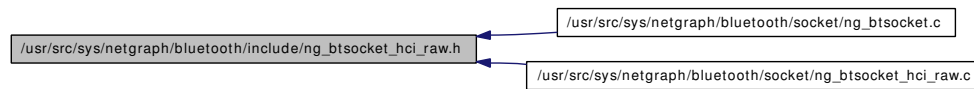
7.44.1.51 #define SOL_RFCOMM 0x0816

Definition at line 319 of file ng_btsocket.h.

Referenced by ng_btsocket_rfcomm_ctloutput().

7.45 /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_hci_raw.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_btsocket_hci_raw_pcb](#)

Defines

- #define [NG_BT_SOCKET_HCI_RAW_SENDSIZE](#) (4 * 1024)
- #define [NG_BT_SOCKET_HCI_RAW_RECVSPACE](#) (4 * 1024)
- #define [NG_BT_SOCKET_HCI_RAW_DIRECTION](#) (1 << 0)
- #define [NG_BT_SOCKET_HCI_RAW_PRIVILEGED](#) (1 << 1)
- #define [so2hci_raw_pcb\(so\)](#) ((struct [ng_btsocket_hci_raw_pcb](#) *)((so) → so_pcb))

Typedefs

- typedef [ng_btsocket_hci_raw_pcb](#) [ng_btsocket_hci_raw_pcb_t](#)
- typedef [ng_btsocket_hci_raw_pcb](#) * [ng_btsocket_hci_raw_pcb_p](#)

Functions

- void [ng_btsocket_hci_raw_init](#) (void)
- void [ng_btsocket_hci_raw_abort](#) (struct socket *)
- void [ng_btsocket_hci_raw_close](#) (struct socket *)
- int [ng_btsocket_hci_raw_attach](#) (struct socket *, int, struct thread *)
- int [ng_btsocket_hci_raw_bind](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_hci_raw_connect](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_hci_raw_control](#) (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)
- int [ng_btsocket_hci_raw_ctloutput](#) (struct socket *, struct sockopt *)
- void [ng_btsocket_hci_raw_detach](#) (struct socket *)
- int [ng_btsocket_hci_raw_disconnect](#) (struct socket *)
- int [ng_btsocket_hci_raw_peeraddr](#) (struct socket *, struct sockaddr **)
- int [ng_btsocket_hci_raw_send](#) (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)
- int [ng_btsocket_hci_raw_sockaddr](#) (struct socket *, struct sockaddr **)

7.45.1 Define Documentation

7.45.1.1 `#define NG_BT_SOCKET_HCI_RAW_DIRECTION (1 << 0)`

Definition at line 47 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_ctloutput()`, and `ng_btsocket_hci_raw_savctl()`.

7.45.1.2 `#define NG_BT_SOCKET_HCI_RAW_PRIVILEGED (1 << 1)`

Definition at line 48 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_attach()`, `ng_btsocket_hci_raw_control()`, and `ng_btsocket_hci_raw_filter()`.

7.45.1.3 `#define NG_BT_SOCKET_HCI_RAW_RECVSPACE (4 * 1024)`

Definition at line 38 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_attach()`.

7.45.1.4 `#define NG_BT_SOCKET_HCI_RAW_SENDSIZE (4 * 1024)`

Definition at line 37 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_attach()`.

7.45.1.5 `#define so2hci_raw_pcb(so) ((struct ng_btsocket_hci_raw_pcb *)((so) → so_pcb))`

Definition at line 59 of file `ng_btsocket_hci_raw.h`.

Referenced by `ng_btsocket_hci_raw_attach()`, `ng_btsocket_hci_raw_bind()`, `ng_btsocket_hci_raw_connect()`, `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_ctloutput()`, `ng_btsocket_hci_raw_detach()`, `ng_btsocket_hci_raw_disconnect()`, `ng_btsocket_hci_raw_send()`, and `ng_btsocket_hci_raw_sockaddr()`.

7.45.2 Typedef Documentation

7.45.2.1 `typedef struct ng_btsocket_hci_raw_pcb* ng_btsocket_hci_raw_pcb_p`

Definition at line 57 of file `ng_btsocket_hci_raw.h`.

7.45.2.2 `typedef struct ng_btsocket_hci_raw_pcb ng_btsocket_hci_raw_pcb_t`

Definition at line 56 of file `ng_btsocket_hci_raw.h`.

7.45.3 Function Documentation

7.45.3.1 `void ng_btsocket_hci_raw_abort (struct socket *)`

Definition at line 878 of file `ng_btsocket_hci_raw.c`.

7.45.3.2 int ng_btsocket_hci_raw_attach (struct socket *, int, struct thread *)

Definition at line 892 of file ng_btsocket_hci_raw.c.

References BLUETOOTH_PROTO_HCI, M_NETGRAPH_BT_SOCKET_HCI_RAW, ng_btsocket_hci_raw_node, NG_BT_SOCKET_HCI_RAW_PRIVILEGED, NG_BT_SOCKET_HCI_RAW_RECVSPACE, NG_BT_SOCKET_HCI_RAW_SENDSpace, NG_HCI_EVENT_COMMAND_COMPL, NG_HCI_EVENT_COMMAND_STATUS, and so2hci_raw_pcb.

7.45.3.3 int ng_btsocket_hci_raw_bind (struct socket *, struct sockaddr *, struct thread *)

Definition at line 945 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::addr, sockaddr_hci::hci_family, sockaddr_hci::hci_len, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

7.45.3.4 void ng_btsocket_hci_raw_close (struct socket *)

Definition at line 883 of file ng_btsocket_hci_raw.c.

7.45.3.5 int ng_btsocket_hci_raw_connect (struct socket *, struct sockaddr *, struct thread *)

Definition at line 977 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::addr, sockaddr_hci::hci_family, sockaddr_hci::hci_len, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

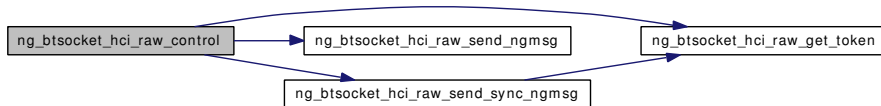
7.45.3.6 int ng_btsocket_hci_raw_control (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)

Definition at line 1016 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::addr, ng_btsocket_hci_raw_node_bdaddr::bdaddr, ng_btsocket_hci_raw_node_buffer::buffer, ng_msg::ng_msghdr::cmd, ng_btsocket_hci_raw_con_list::connections, ng_msg::data, ng_btsocket_hci_raw_node_debug::debug, ng_btsocket_hci_raw_node_neighbor_cache::entries, ng_btsocket_hci_raw_node_features::features, ng_btsocket_hci_raw_pcb::flags, sockaddr_hci::hci_node, ng_msg::header, min, ng_btsocket_hci_raw_pcb::msg, ng_btsocket_hci_raw_node_list_names::names, ng_btsocket_hci_raw_get_token(), ng_btsocket_hci_raw_node, NG_BT_SOCKET_HCI_RAW_PRIVILEGED, ng_btsocket_hci_raw_send_ngmsg(), ng_btsocket_hci_raw_send_sync_ngmsg(), NG_FREE_MSG, NG_HCI_MAX_CON_NUM, NG_HCI_MAX_NEIGHBOR_NUM, NG_HCI_NODE_TYPE, NG_MKMESSAGE, NG_NODESIZ, NG_SEND_MSG_PATH, NGM_GENERIC_COOKIE, NGM_HCI_COOKIE, NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE, NGM_HCI_NODE_GET_BDADDR, NGM_HCI_NODE_GET_BUFFER, NGM_HCI_NODE_GET_CON_LIST, NGM_HCI_NODE_GET_DEBUG, NGM_HCI_NODE_GET_FEATURES, NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK, NGM_HCI_NODE_GET_NEIGHBOR_CACHE, NGM_HCI_NODE_GET_PACKET_MASK, NGM_HCI_NODE_GET_ROLE_SWITCH, NGM_HCI_NODE_GET_STAT, NGM_HCI_NODE_GET_STATE, NGM_HCI_NODE_INIT, NGM_HCI_NODE_RESET_STAT, NGM_HCI_NODE_SET_DEBUG, NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK, NGM_HCI_NODE_SET_PACKET_MASK, NGM_HCI_NODE_SET_ROLE_SWITCH, NGM_LISTNAMES, namelist::nodeinfo, ng_btsocket_hci_raw_con_list::num_connections, ng_hci_node_con_list_ep::num_connections, ng_btsocket_hci_raw_node_neighbor_cache::num_entries, ng_hci_node_get_neighbor_cache_ep::num_entries, ng_btsocket_hci_raw_node_list_names::num_names, namelist::numnames, ng_btsocket_hci_raw_node_packet_mask::packet_mask, ng_btsocket_

hci_raw_pcb::pcb_mtx, ng_btsocket_hci_raw_node_link_policy_mask::policy_mask, ng_btsocket_hci_raw_node_role_switch::role_switch, SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_CACHE, SIOC_HCI_RAW_NODE_GET_BDADDR, SIOC_HCI_RAW_NODE_GET_BUFFER, SIOC_HCI_RAW_NODE_GET_CON_LIST, SIOC_HCI_RAW_NODE_GET_DEBUG, SIOC_HCI_RAW_NODE_GET_FEATURES, SIOC_HCI_RAW_NODE_GET_LINK_POLICY_MASK, SIOC_HCI_RAW_NODE_GET_NEIGHBOR_CACHE, SIOC_HCI_RAW_NODE_GET_PACKET_MASK, SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH, SIOC_HCI_RAW_NODE_GET_STAT, SIOC_HCI_RAW_NODE_GET_STATE, SIOC_HCI_RAW_NODE_INIT, SIOC_HCI_RAW_NODE_LIST_NAMES, SIOC_HCI_RAW_NODE_RESET_STAT, SIOC_HCI_RAW_NODE_SET_DEBUG, SIOC_HCI_RAW_NODE_SET_LINK_POLICY_MASK, SIOC_HCI_RAW_NODE_SET_PACKET_MASK, SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH, so2hci_raw_pcb, ng_btsocket_hci_raw_node_state::stat, ng_btsocket_hci_raw_node_state::state, ng_btsocket_hci_raw_pcb::token, and ng_mesg::ng_msghdr::token.

Here is the call graph for this function:



7.45.3.7 int ng_btsocket_hci_raw_ctloutput (struct socket *, struct sockopt *)

Definition at line 1393 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::filter, ng_btsocket_hci_raw_pcb::flags, NG_BT_SOCKET_HCI_RAW_DIRECTION, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, so2hci_raw_pcb, SO_HCI_RAW_DIRECTION, SO_HCI_RAW_FILTER, and SOL_HCI_RAW.

7.45.3.8 void ng_btsocket_hci_raw_detach (struct socket *)

Definition at line 1471 of file ng_btsocket_hci_raw.c.

References M_NETGRAPH_BT_SOCKET_HCI_RAW, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

7.45.3.9 int ng_btsocket_hci_raw_disconnect (struct socket *)

Definition at line 1501 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

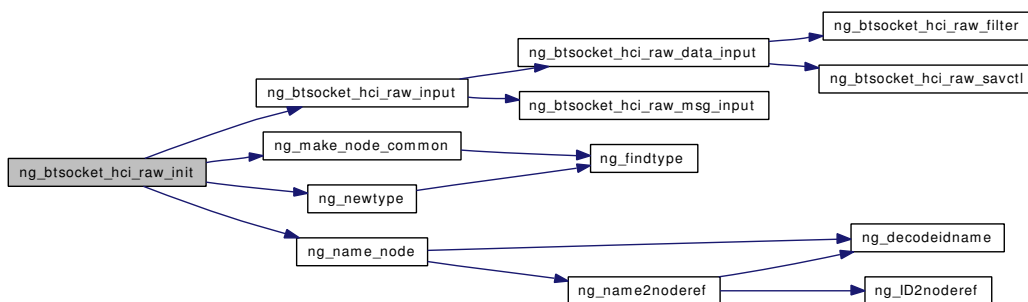
7.45.3.10 void ng_btsocket_hci_raw_init (void)

Definition at line 721 of file ng_btsocket_hci_raw.c.

References ifqmaxlen, M_NETGRAPH_BT_SOCKET_HCI_RAW, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_HCI_RAW_ALERT, ng_btsocket_hci_raw_input(), ng_btsocket_hci_raw_node, NG_BT_SOCKET_HCI_RAW_NODE_TYPE, ng_btsocket_hci_raw_queue, ng_btsocket_hci_raw_queue_mtx, ng_btsocket_hci_raw_task, NG_HCI_EVENT_LINK_KEY_NOTIFICATION, NG_HCI_EVENT_RETURN_LINK_KEYS, NG_HCI_EVENT_VENDOR, NG_HCI_OCF_EXIT_PERIODIC_INQUIRY,

NG_HCI_OCF_GET_LINK_QUALITY, NG_HCI_OCF_INQUIRY, NG_HCI_OCF_INQUIRY_CANCEL, NG_HCI_OCF_PERIODIC_INQUIRY, NG_HCI_OCF_READ_AUTH_ENABLE, NG_HCI_OCF_READ_AUTO_FLUSH_TIMO, NG_HCI_OCF_READ_BDADDR, NG_HCI_OCF_READ_BUFFER_SIZE, NG_HCI_OCF_READ_CLOCK_OFFSET, NG_HCI_OCF_READ_CON_ACCEPT_TIMO, NG_HCI_OCF_READ_COUNTRY_CODE, NG_HCI_OCF_READ_ENCRYPTION_MODE, NG_HCI_OCF_READ_FAILED_CONTACT_CNTR, NG_HCI_OCF_READ_HOLD_MODE_ACTIVITY, NG_HCI_OCF_READ_IAC_LAP, NG_HCI_OCF_READ_INQUIRY_SCAN_ACTIVITY, NG_HCI_OCF_READ_LINK_POLICY_SETTINGS, NG_HCI_OCF_READ_LINK_SUPERVISION_TIMO, NG_HCI_OCF_READ_LOCAL_FEATURES, NG_HCI_OCF_READ_LOCAL_NAME, NG_HCI_OCF_READ_LOCAL_VER, NG_HCI_OCF_READ_LOOPBACK_MODE, NG_HCI_OCF_READ_NUM_BROADCAST_RETRANS, NG_HCI_OCF_READ_PAGE_SCAN, NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY, NG_HCI_OCF_READ_PAGE_SCAN_PERIOD, NG_HCI_OCF_READ_PAGE_TIMO, NG_HCI_OCF_READ_PIN_TYPE, NG_HCI_OCF_READ_REMOTE_FEATURES, NG_HCI_OCF_READ_REMOTE_VER_INFO, NG_HCI_OCF_READ_RSSI, NG_HCI_OCF_READ_SCAN_ENABLE, NG_HCI_OCF_READ_SCO_FLOW_CONTROL, NG_HCI_OCF_READ_SUPPORTED_IAC_NUM, NG_HCI_OCF_READ_UNIT_CLASS, NG_HCI_OCF_READ_VOICE_SETTINGS, NG_HCI_OCF_READ_XMIT_LEVEL, NG_HCI_OCF_REMOTE_NAME_REQ, NG_HCI_OCF_ROLE_DISCOVERY, NG_HCI_OGF_HC_BASEBAND, NG_HCI_OGF_INFO, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OGF_LINK_POLICY, NG_HCI_OGF_STATUS, NG_HCI_OGF_TESTING, ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



7.45.3.11 int ng_btsocket_hci_raw_peeraddr (struct socket *, struct sockaddr **)

Definition at line 1522 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_sockaddr().

Here is the call graph for this function:

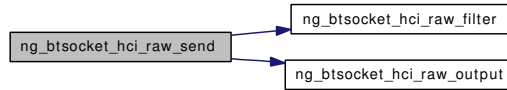


7.45.3.12 int ng_btsocket_hci_raw_send (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)

Definition at line 1532 of file ng_btsocket_hci_raw.c.

References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_node`, `ng_btsocket_hci_raw_filter()`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_output()`, `NG_FREE_M`, `NG_HCI_CMD_PKT`, `NG_HCI_CMD_PKT_SIZE`, `ng_send_fn`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

Here is the call graph for this function:



7.45.3.13 `int ng_btsocket_hci_raw_sockaddr (struct socket *, struct sockaddr **)`

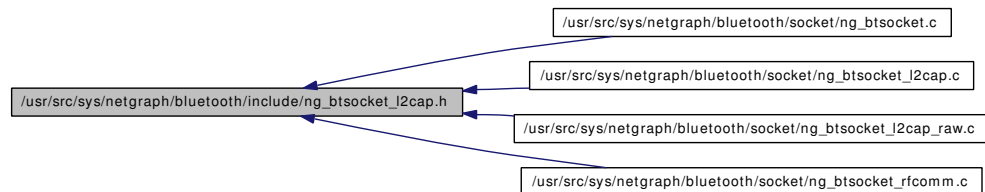
Definition at line 1617 of file `ng_btsocket_hci_raw.c`.

References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_node`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

Referenced by `ng_btsocket_hci_raw_peeraddr()`.

7.46 /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_btsocket_l2cap_rtenry](#)
- struct [ng_btsocket_l2cap_raw_pcb](#)
- struct [ng_btsocket_l2cap_pcb](#)

Defines

- #define [NG_BT_SOCKET_L2CAP_RAW_SENDSPACE](#) [NG_L2CAP_MTU_DEFAULT](#)
- #define [NG_BT_SOCKET_L2CAP_RAW_RECVSPACE](#) [NG_L2CAP_MTU_DEFAULT](#)
- #define [NG_BT_SOCKET_L2CAP_RAW_PRIVILEGED](#) (1 << 0)
- #define [so2l2cap_raw_pcb\(so\)](#) ((struct [ng_btsocket_l2cap_raw_pcb](#) *)((so) → so_pcb))
- #define [NG_BT_SOCKET_L2CAP_SENDSPACE](#) [NG_L2CAP_MTU_DEFAULT](#)
- #define [NG_BT_SOCKET_L2CAP_RECVSPACE](#) (64 * 1024)
- #define [NG_BT_SOCKET_L2CAP_CLIENT](#) (1 << 0)
- #define [NG_BT_SOCKET_L2CAP_TIMO](#) (1 << 1)
- #define [NG_BT_SOCKET_L2CAP_CLOSED](#) 0
- #define [NG_BT_SOCKET_L2CAP_CONNECTING](#) 1
- #define [NG_BT_SOCKET_L2CAP_CONFIGURING](#) 2
- #define [NG_BT_SOCKET_L2CAP_OPEN](#) 3
- #define [NG_BT_SOCKET_L2CAP_DISCONNECTING](#) 4
- #define [NG_BT_SOCKET_L2CAP_CFG_IN](#) (1 << 0)
- #define [NG_BT_SOCKET_L2CAP_CFG_OUT](#) (1 << 1)
- #define [NG_BT_SOCKET_L2CAP_CFG_BOTH](#) (NG_BT_SOCKET_L2CAP_CFG_IN | NG_BT_SOCKET_L2CAP_CFG_OUT)
- #define [NG_BT_SOCKET_L2CAP_CFG_IN_SENT](#) (1 << 2)
- #define [NG_BT_SOCKET_L2CAP_CFG_OUT_SENT](#) (1 << 3)
- #define [so2l2cap_pcb\(so\)](#) ((struct [ng_btsocket_l2cap_pcb](#) *)((so) → so_pcb))

Typedefs

- typedef [ng_btsocket_l2cap_rtenry](#) [ng_btsocket_l2cap_rtenry_t](#)
- typedef [ng_btsocket_l2cap_rtenry](#) * [ng_btsocket_l2cap_rtenry_p](#)
- typedef [ng_btsocket_l2cap_raw_pcb](#) [ng_btsocket_l2cap_raw_pcb_t](#)
- typedef [ng_btsocket_l2cap_raw_pcb](#) * [ng_btsocket_l2cap_raw_pcb_p](#)
- typedef [ng_btsocket_l2cap_pcb](#) [ng_btsocket_l2cap_pcb_t](#)
- typedef [ng_btsocket_l2cap_pcb](#) * [ng_btsocket_l2cap_pcb_p](#)

Functions

- void [ng_btsocket_l2cap_raw_init](#) (void)
- void [ng_btsocket_l2cap_raw_abort](#) (struct socket *)
- void [ng_btsocket_l2cap_raw_close](#) (struct socket *)
- int [ng_btsocket_l2cap_raw_attach](#) (struct socket *, int, struct thread *)
- int [ng_btsocket_l2cap_raw_bind](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_l2cap_raw_connect](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_l2cap_raw_control](#) (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)
- void [ng_btsocket_l2cap_raw_detach](#) (struct socket *)
- int [ng_btsocket_l2cap_raw_disconnect](#) (struct socket *)
- int [ng_btsocket_l2cap_raw_peeraddr](#) (struct socket *, struct sockaddr **)
- int [ng_btsocket_l2cap_raw_send](#) (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)
- int [ng_btsocket_l2cap_raw_sockaddr](#) (struct socket *, struct sockaddr **)
- void [ng_btsocket_l2cap_init](#) (void)
- void [ng_btsocket_l2cap_abort](#) (struct socket *)
- void [ng_btsocket_l2cap_close](#) (struct socket *)
- int [ng_btsocket_l2cap_accept](#) (struct socket *, struct sockaddr **)
- int [ng_btsocket_l2cap_attach](#) (struct socket *, int, struct thread *)
- int [ng_btsocket_l2cap_bind](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_l2cap_connect](#) (struct socket *, struct sockaddr *, struct thread *)
- int [ng_btsocket_l2cap_control](#) (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)
- int [ng_btsocket_l2cap_ctloutput](#) (struct socket *, struct sockopt *)
- void [ng_btsocket_l2cap_detach](#) (struct socket *)
- int [ng_btsocket_l2cap_disconnect](#) (struct socket *)
- int [ng_btsocket_l2cap_listen](#) (struct socket *, int, struct thread *)
- int [ng_btsocket_l2cap_peeraddr](#) (struct socket *, struct sockaddr **)
- int [ng_btsocket_l2cap_send](#) (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)
- int [ng_btsocket_l2cap_sockaddr](#) (struct socket *, struct sockaddr **)

7.46.1 Define Documentation

7.46.1.1 #define NG_BT_SOCKET_L2CAP_CFG_BOTH (NG_BT_SOCKET_L2CAP_CFG_IN | NG_BT_SOCKET_L2CAP_CFG_OUT)

Definition at line 150 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, and `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`.

7.46.1.2 #define NG_BT_SOCKET_L2CAP_CFG_IN (1 << 0)

Definition at line 148 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`.

7.46.1.3 #define NG_BT_SOCKET_L2CAP_CFG_IN_SENT (1 << 2)

Definition at line 153 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), and ng_btsocket_l2cap_process_l2ca_con_req_rsp().

7.46.1.4 #define NG_BT_SOCKET_L2CAP_CFG_OUT (1 << 1)

Definition at line 149 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp().

7.46.1.5 #define NG_BT_SOCKET_L2CAP_CFG_OUT_SENT (1 << 3)

Definition at line 154 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_cfg_ind(), and ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp().

7.46.1.6 #define NG_BT_SOCKET_L2CAP_CLIENT (1 << 0)

Definition at line 137 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_connect(), and ng_btsocket_l2cap_process_l2ca_con_ind().

7.46.1.7 #define NG_BT_SOCKET_L2CAP_CLOSED 0

Definition at line 141 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_attach(), ng_btsocket_l2cap_ctloutput(), ng_btsocket_l2cap_detach(), ng_btsocket_l2cap_disconnect(), ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_discon_ind(), ng_btsocket_l2cap_process_l2ca_discon_rsp(), ng_btsocket_l2cap_process_timeout(), and ng_btsocket_l2cap_rtclean().

7.46.1.8 #define NG_BT_SOCKET_L2CAP_CONFIGURING 2

Definition at line 143 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), and ng_btsocket_l2cap_process_timeout().

7.46.1.9 #define NG_BT_SOCKET_L2CAP_CONNECTING 1

Definition at line 142 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_connect(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), and ng_btsocket_l2cap_process_timeout().

7.46.1.10 #define NG_BTSOCKET_L2CAP_DISCONNECTING 4

Definition at line 145 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_detach(), ng_btsocket_l2cap_disconnect(), and ng_btsocket_l2cap_process_timeout().

7.46.1.11 #define NG_BTSOCKET_L2CAP_OPEN 3

Definition at line 144 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_write_rsp(), ng_btsocket_l2cap_process_timeout(), and ng_btsocket_l2cap_send().

7.46.1.12 #define NG_BTSOCKET_L2CAP_RAW_PRIVILEGED (1 << 0)

Definition at line 69 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_attach(), and ng_btsocket_l2cap_raw_control().

7.46.1.13 #define NG_BTSOCKET_L2CAP_RAW_RECVSPACE NG_L2CAP_MTU_DEFAULT

Definition at line 59 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_attach().

7.46.1.14 #define NG_BTSOCKET_L2CAP_RAW_SENDSPACE NG_L2CAP_MTU_DEFAULT

Definition at line 58 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_attach().

7.46.1.15 #define NG_BTSOCKET_L2CAP_RECVSPACE (64 * 1024)

Definition at line 121 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_attach().

7.46.1.16 #define NG_BTSOCKET_L2CAP_SENDSPACE NG_L2CAP_MTU_DEFAULT

Definition at line 120 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_attach().

7.46.1.17 #define NG_BTSOCKET_L2CAP_TIMO (1 << 1)

Definition at line 138 of file ng_btsocket_l2cap.h.

Referenced by ng_btsocket_l2cap_detach(), ng_btsocket_l2cap_disconnect(), ng_btsocket_l2cap_process_l2ca_discon_ind(), ng_btsocket_l2cap_process_timeout(), ng_btsocket_l2cap_rtclean(), ng_btsocket_l2cap_send(), ng_btsocket_l2cap_timeout(), and ng_btsocket_l2cap_untimeout().

7.46.1.18 `#define so2l2cap_pcb(so) ((struct ng_btsocket_l2cap_pcb *)((so) → so_pcb))`

Definition at line 177 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_attach()`, `ng_btsocket_l2cap_bind()`, `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_ctloutput()`, `ng_btsocket_l2cap_detach()`, `ng_btsocket_l2cap_disconnect()`, `ng_btsocket_l2cap_listen()`, `ng_btsocket_l2cap_peeraddr()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_send()`, `ng_btsocket_l2cap_sockaddr()`, `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_connect_ind()`, `ng_btsocket_rfcomm_session_accept()`, `ng_btsocket_rfcomm_session_by_addr()`, and `ng_btsocket_rfcomm_session_connect()`.

7.46.1.19 `#define so2l2cap_raw_pcb(so) ((struct ng_btsocket_l2cap_raw_pcb *)((so) → so_pcb))`

Definition at line 85 of file `ng_btsocket_l2cap.h`.

Referenced by `ng_btsocket_l2cap_raw_attach()`, `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_detach()`, `ng_btsocket_l2cap_raw_disconnect()`, `ng_btsocket_l2cap_raw_peeraddr()`, and `ng_btsocket_l2cap_raw_sockaddr()`.

7.46.2 Typedef Documentation

7.46.2.1 `typedef struct ng_btsocket_l2cap_pcb* ng_btsocket_l2cap_pcb_p`

Definition at line 175 of file `ng_btsocket_l2cap.h`.

7.46.2.2 `typedef struct ng_btsocket_l2cap_pcb ng_btsocket_l2cap_pcb_t`

Definition at line 174 of file `ng_btsocket_l2cap.h`.

7.46.2.3 `typedef struct ng_btsocket_l2cap_raw_pcb* ng_btsocket_l2cap_raw_pcb_p`

Definition at line 83 of file `ng_btsocket_l2cap.h`.

7.46.2.4 `typedef struct ng_btsocket_l2cap_raw_pcb ng_btsocket_l2cap_raw_pcb_t`

Definition at line 82 of file `ng_btsocket_l2cap.h`.

7.46.2.5 `typedef struct ng_btsocket_l2cap_rtrentry* ng_btsocket_l2cap_rtrentry_p`

Definition at line 50 of file `ng_btsocket_l2cap.h`.

7.46.2.6 `typedef struct ng_btsocket_l2cap_rtrentry ng_btsocket_l2cap_rtrentry_t`

Definition at line 49 of file `ng_btsocket_l2cap.h`.

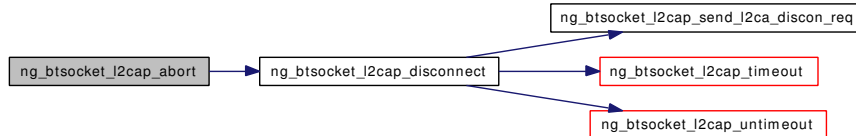
7.46.3 Function Documentation

7.46.3.1 void ng_btsocket_l2cap_abort (struct socket *)

Definition at line 1864 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_disconnect().

Here is the call graph for this function:



7.46.3.2 int ng_btsocket_l2cap_accept (struct socket *, struct sockaddr **)

Definition at line 1884 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_node, and ng_btsocket_l2cap_peeraddr().

Here is the call graph for this function:



7.46.3.3 int ng_btsocket_l2cap_attach (struct socket *, int, struct thread *)

Definition at line 1897 of file ng_btsocket_l2cap.c.

References BLUETOOTH_PROTO_L2CAP, M_NETGRAPH_BT SOCKET_L2CAP, NG_BT SOCKET_L2CAP_CLOSED, ng_btsocket_l2cap_node, NG_BT SOCKET_L2CAP_RECVSPACE, NG_BT SOCKET_L2CAP_SENDSpace, NG_HCI_SERVICE_TYPE_BEST_EFFORT, NG_L2CAP_FLUSH_TIMO_DEFAULT, NG_L2CAP_LINK_TIMO_DEFAULT, NG_L2CAP_MTU_DEFAULT, and so2l2cap_pcb.

7.46.3.4 int ng_btsocket_l2cap_bind (struct socket *, struct sockaddr *, struct thread *)

Definition at line 2009 of file ng_btsocket_l2cap.c.

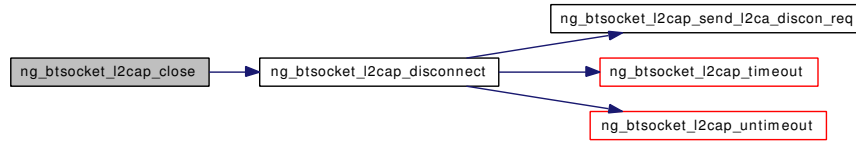
References sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::src.

7.46.3.5 void ng_btsocket_l2cap_close (struct socket *)

Definition at line 1872 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_disconnect().

Here is the call graph for this function:

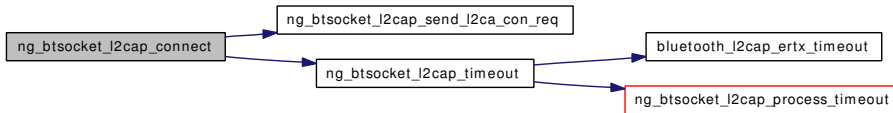


7.46.3.6 int ng_btsocket_l2cap_connect (struct socket *, struct sockaddr *, struct thread *)

Definition at line 2065 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::dst, ng_btsocket_l2cap_pcb::flags, ng_btsocket_l2cap_rtenry::hook, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, NG_BTsocket_L2CAP_CLIENT, NG_BTsocket_L2CAP_CONNECTING, ng_btsocket_l2cap_node, ng_btsocket_l2cap_send_l2ca_con_req(), ng_btsocket_l2cap_timeout(), NG_HCI_BDADDR_ANY, NG_HOOK_NOT_VALID, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::rt, ng_btsocket_l2cap_pcb::so, so2l2cap_pcb, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_rtenry::src, and ng_btsocket_l2cap_pcb::state.

Here is the call graph for this function:



7.46.3.7 int ng_btsocket_l2cap_control (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)

Definition at line 2161 of file ng_btsocket_l2cap.c.

7.46.3.8 int ng_btsocket_l2cap_ctloutput (struct socket *, struct sockopt *)

Definition at line 2172 of file ng_btsocket_l2cap.c.

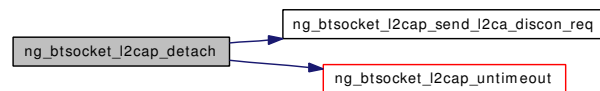
References ng_btsocket_l2cap_pcb::flush_timo, ng_btsocket_l2cap_pcb::iflow, ng_btsocket_l2cap_pcb::imtu, NG_BTsocket_L2CAP_CLOSED, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::oflow, ng_btsocket_l2cap_pcb::omtu, ng_btsocket_l2cap_pcb::pcb_mtx, so2l2cap_pcb, SO_L2CAP_FLUSH, SO_L2CAP_IFLOW, SO_L2CAP_IMTU, SO_L2CAP_OFLOW, SO_L2CAP_OMTU, SOL_L2CAP, and ng_btsocket_l2cap_pcb::state.

7.46.3.9 void ng_btsocket_l2cap_detach (struct socket *)

Definition at line 2275 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::flags, M_NETGRAPH_BTsocket_L2CAP, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_DISCONNECTING, ng_btsocket_l2cap_node, ng_btsocket_l2cap_send_l2ca_discon_req(), NG_BTsocket_L2CAP_TIMO, ng_btsocket_l2cap_untimeout(), ng_btsocket_l2cap_pcb::pcb_mtx, and so2l2cap_pcb.

Here is the call graph for this function:



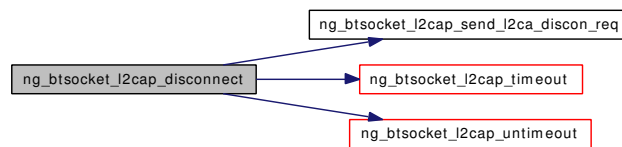
7.46.3.10 int ng_btsocket_l2cap_disconnect (struct socket *)

Definition at line 2316 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::flags, NG_BT_SOCKET_L2CAP_CLOSED, NG_BT_SOCKET_L2CAP_DISCONNECTING, ng_btsocket_l2cap_node, ng_btsocket_l2cap_send_l2ca_discon_req(), ng_btsocket_l2cap_timeout(), NG_BT_SOCKET_L2CAP_TIMO, ng_btsocket_l2cap_untimeout(), ng_btsocket_l2cap_pcb::pcb_mtx, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::state.

Referenced by ng_btsocket_l2cap_abort(), and ng_btsocket_l2cap_close().

Here is the call graph for this function:

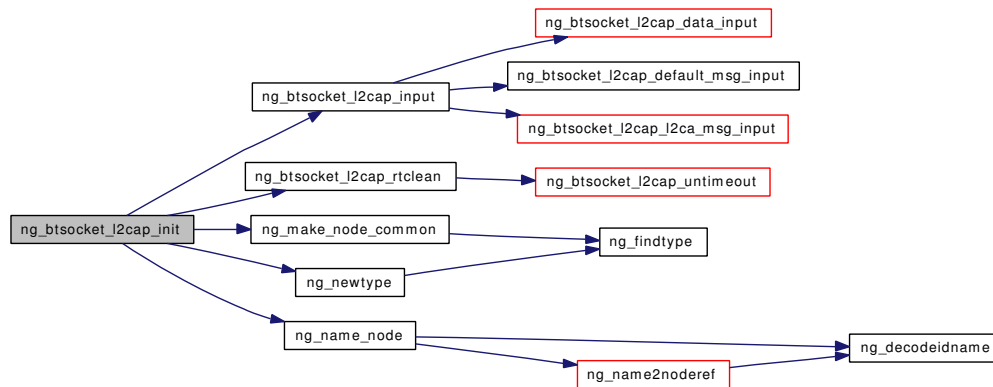


7.46.3.11 void ng_btsocket_l2cap_init (void)

Definition at line 1800 of file ng_btsocket_l2cap.c.

References ifqmaxlen, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_L2CAP_ALERT, ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node, NG_BT_SOCKET_L2CAP_NODE_TYPE, ng_btsocket_l2cap_queue, ng_btsocket_l2cap_queue_mtx, ng_btsocket_l2cap_queue_task, ng_btsocket_l2cap_rtclean(), ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



7.46.3.12 int ng_btsocket_l2cap_listen (struct socket *, int, struct thread *)

Definition at line 2359 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, and so2l2cap_pcb.

7.46.3.13 int ng_btsocket_l2cap_peeraddr (struct socket *, struct sockaddr **)

Definition at line 2391 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::dst, sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, and so2l2cap_pcb.

Referenced by ng_btsocket_l2cap_accept().

7.46.3.14 void ng_btsocket_l2cap_raw_abort (struct socket *)

Definition at line 577 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_disconnect().

Here is the call graph for this function:

**7.46.3.15 int ng_btsocket_l2cap_raw_attach (struct socket *, int, struct thread *)**

Definition at line 595 of file ng_btsocket_l2cap_raw.c.

References M_NETGRAPH_BT_SOCKET_L2CAP_RAW, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_PRIVILEGED, NG_BT_SOCKET_L2CAP_RAW_RECVSPACE, NG_BT_SOCKET_L2CAP_RAW_SENDSPEACE, and so2l2cap_raw_pcb.

7.46.3.16 int ng_btsocket_l2cap_raw_bind (struct socket *, struct sockaddr *, struct thread *)

Definition at line 642 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_rentry::hook, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, ng_btsocket_l2cap_raw_node, NG_HCI_BDADDR_ANY, NG_HOOK_NOT_VALID, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, so2l2cap_raw_pcb, ng_btsocket_l2cap_rentry::src, and ng_btsocket_l2cap_raw_pcb::src.

7.46.3.17 void ng_btsocket_l2cap_raw_close (struct socket *)

Definition at line 584 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_disconnect().

Here is the call graph for this function:



7.46.3.18 `int ng_btsocket_l2cap_raw_connect (struct socket *, struct sockaddr *, struct thread *)`

Definition at line 694 of file `ng_btsocket_l2cap_raw.c`.

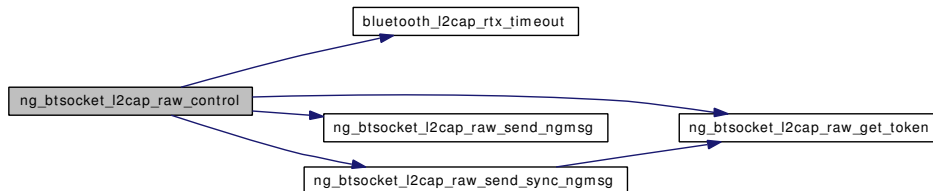
References `ng_btsocket_l2cap_raw_pcb::dst`, `ng_btsocket_l2cap_rtrentry::hook`, `sockaddr_l2cap::l2cap_family`, `sockaddr_l2cap::l2cap_len`, `ng_btsocket_l2cap_raw_node`, `NG_HCI_BDADDR_ANY`, `NG_HOOK_NOT_VALID`, `ng_btsocket_l2cap_raw_pcb::pcb_mtx`, `ng_btsocket_l2cap_raw_pcb::rt`, `so2l2cap_raw_pcb`, `ng_btsocket_l2cap_raw_pcb::src`, and `ng_btsocket_l2cap_rtrentry::src`.

7.46.3.19 `int ng_btsocket_l2cap_raw_control (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)`

Definition at line 772 of file `ng_btsocket_l2cap_raw.c`.

References `bluetooth_l2cap_rtx_timeout()`, `ng_btsocket_l2cap_raw_chan_list::channels`, `ng_mesg::ng_msghdr::cmd`, `ng_btsocket_l2cap_raw_con_list::connections`, `ng_mesg::data`, `ng_btsocket_l2cap_raw_node_debug::debug`, `ng_btsocket_l2cap_raw_pcb::dst`, `ng_btsocket_l2cap_raw_ping::echo_data`, `ng_btsocket_l2cap_raw_ping::echo_size`, `ng_l2cap_l2ca_ping_op::echo_size`, `ng_btsocket_l2cap_raw_node_flags::flags`, `ng_btsocket_l2cap_raw_pcb::flags`, `ng_mesg::header`, `ng_btsocket_l2cap_rtrentry::hook`, `ng_btsocket_l2cap_raw_get_info::info_data`, `ng_btsocket_l2cap_raw_get_info::info_size`, `ng_l2cap_l2ca_get_info_op::info_size`, `ng_btsocket_l2cap_raw_get_info::info_type`, `min`, `ng_btsocket_l2cap_raw_pcb::msg`, `ng_btsocket_l2cap_raw_get_token()`, `ng_btsocket_l2cap_raw_ioctl_timeout`, `ng_btsocket_l2cap_raw_node`, `NG_BTSOCKET_L2CAP_RAW_PRIVILEGED`, `ng_btsocket_l2cap_raw_send_ngmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `NG_FREE_MSG`, `NG_L2CAP_MAX_CHAN_NUM`, `NG_L2CAP_MAX_CON_NUM`, `NG_L2CAP_MAX_ECHO_SIZE`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_GET_INFO`, `NGM_L2CAP_L2CA_PING`, `NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO`, `NGM_L2CAP_NODE_GET_CHAN_LIST`, `NGM_L2CAP_NODE_GET_CON_LIST`, `NGM_L2CAP_NODE_GET_DEBUG`, `NGM_L2CAP_NODE_GET_FLAGS`, `NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO`, `NGM_L2CAP_NODE_SET_DEBUG`, `ng_btsocket_l2cap_raw_chan_list::num_channels`, `ng_l2cap_node_chan_list_ep::num_channels`, `ng_btsocket_l2cap_raw_con_list::num_connections`, `ng_l2cap_node_con_list_ep::num_connections`, `ng_btsocket_l2cap_raw_pcb::pcb_mtx`, `ng_btsocket_l2cap_raw_ping::result`, `ng_l2cap_l2ca_ping_op::result`, `ng_btsocket_l2cap_raw_get_info::result`, `ng_l2cap_l2ca_get_info_op::result`, `ng_btsocket_l2cap_raw_pcb::rt`, `SIOC_L2CAP_L2CA_GET_INFO`, `SIOC_L2CAP_L2CA_PING`, `SIOC_L2CAP_NODE_GET_AUTO_DISCON_TIMO`, `SIOC_L2CAP_NODE_GET_CHAN_LIST`, `SIOC_L2CAP_NODE_GET_CON_LIST`, `SIOC_L2CAP_NODE_GET_DEBUG`, `SIOC_L2CAP_NODE_GET_FLAGS`, `SIOC_L2CAP_NODE_SET_AUTO_DISCON_TIMO`, `SIOC_L2CAP_NODE_SET_DEBUG`, `so2l2cap_raw_pcb`, `ng_btsocket_l2cap_raw_auto_discon_timo::timeout`, `ng_btsocket_l2cap_raw_pcb::token`, and `ng_mesg::ng_msghdr::token`.

Here is the call graph for this function:



7.46.3.20 `void ng_btsocket_l2cap_raw_detach (struct socket *)`

Definition at line 1107 of file `ng_btsocket_l2cap_raw.c`.

References M_NETGRAPH_BT_SOCKET_L2CAP_RAW, ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, and so2l2cap_raw_pcb.

7.46.3.21 int ng_btsocket_l2cap_raw_disconnect (struct socket *)

Definition at line 1136 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, and so2l2cap_raw_pcb.

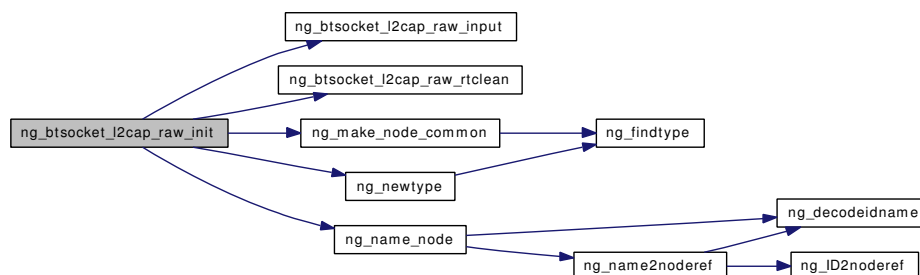
Referenced by ng_btsocket_l2cap_raw_abort(), and ng_btsocket_l2cap_raw_close().

7.46.3.22 void ng_btsocket_l2cap_raw_init (void)

Definition at line 507 of file ng_btsocket_l2cap_raw.c.

References ifqmaxlen, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_L2CAP_RAW_ALERT, ng_btsocket_l2cap_raw_debug_level, ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_ioctl_timeout, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE, ng_btsocket_l2cap_raw_queue, ng_btsocket_l2cap_raw_queue_mtx, ng_btsocket_l2cap_raw_queue_task, ng_btsocket_l2cap_raw_rtclean(), NG_BT_SOCKET_WARN_LEVEL, ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



7.46.3.23 int ng_btsocket_l2cap_raw_peeraddr (struct socket *, struct sockaddr **)

Definition at line 1158 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_pcb::dst, sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, and so2l2cap_raw_pcb.

7.46.3.24 int ng_btsocket_l2cap_raw_send (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)

Definition at line 1186 of file ng_btsocket_l2cap_raw.c.

References NG_FREE_M.

7.46.3.25 int ng_btsocket_l2cap_raw_sockaddr (struct socket *, struct sockaddr **)

Definition at line 1200 of file ng_btsocket_l2cap_raw.c.

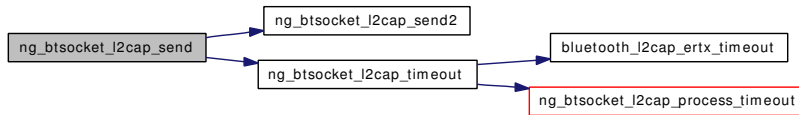
References sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, so2l2cap_raw_pcb, and ng_btsocket_l2cap_raw_pcb::src.

7.46.3.26 int ng_btsocket_l2cap_send (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)

Definition at line 2416 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::flags, ng_btsocket_l2cap_rtentry::hook, NG_BTsocket_L2CAP_ERR, ng_btsocket_l2cap_node, NG_BTsocket_L2CAP_OPEN, ng_btsocket_l2cap_send2(), ng_btsocket_l2cap_timeout(), NG_BTsocket_L2CAP_TIMO, NG_FREE_M, NG_HOOK_NOT_VALID, ng_btsocket_l2cap_pcb::omtu, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::rt, ng_btsocket_l2cap_pcb::so, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::state.

Here is the call graph for this function:



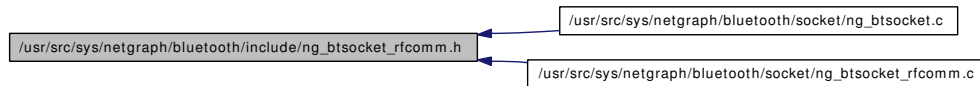
7.46.3.27 int ng_btsocket_l2cap_sockaddr (struct socket *, struct sockaddr **)

Definition at line 2544 of file ng_btsocket_l2cap.c.

References sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::src.

7.47 /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [rfcomm_frame_hdr](#)
- struct [rfcomm_cmd_hdr](#)
- struct [rfcomm_mcc_hdr](#)
- struct [rfcomm_mcc_msc](#)
- struct [rfcomm_mcc_rpn](#)
- struct [rfcomm_mcc_rls](#)
- struct [rfcomm_mcc_pn](#)
- struct [ng_btsocket_rfcomm_session](#)
- struct [ng_btsocket_rfcomm_pcb](#)

Defines

- #define [RFCOMM_DEFAULT_MTU](#) 667
- #define [RFCOMM_MAX_MTU](#) 1024
- #define [RFCOMM_DEFAULT_CREDITS](#) 7
- #define [RFCOMM_MAX_CREDITS](#) 40
- #define [RFCOMM_FRAME_SABM](#) 0x2f
- #define [RFCOMM_FRAME_DISC](#) 0x43
- #define [RFCOMM_FRAME_UA](#) 0x63
- #define [RFCOMM_FRAME_DM](#) 0x0f
- #define [RFCOMM_FRAME_UIH](#) 0xef
- #define [RFCOMM_MCC_TEST](#) 0x08
- #define [RFCOMM_MCC_FCON](#) 0x28
- #define [RFCOMM_MCC_FCOFF](#) 0x18
- #define [RFCOMM_MCC_MSC](#) 0x38
- #define [RFCOMM_MCC_RPN](#) 0x24
- #define [RFCOMM_MCC_RLS](#) 0x14
- #define [RFCOMM_MCC_PN](#) 0x20
- #define [RFCOMM_MCC_NSC](#) 0x04
- #define [RFCOMM_MODEM_FC](#) 0x02
- #define [RFCOMM_MODEM_RTC](#) 0x04
- #define [RFCOMM_MODEM_RTR](#) 0x08
- #define [RFCOMM_MODEM_IC](#) 0x40
- #define [RFCOMM_MODEM_DV](#) 0x80
- #define [RFCOMM_RPN_BR_2400](#) 0x0
- #define [RFCOMM_RPN_BR_4800](#) 0x1
- #define [RFCOMM_RPN_BR_7200](#) 0x2

- #define RFCOMM_RPN_BR_9600 0x3
- #define RFCOMM_RPN_BR_19200 0x4
- #define RFCOMM_RPN_BR_38400 0x5
- #define RFCOMM_RPN_BR_57600 0x6
- #define RFCOMM_RPN_BR_115200 0x7
- #define RFCOMM_RPN_BR_230400 0x8
- #define RFCOMM_RPN_DATA_5 0x0
- #define RFCOMM_RPN_DATA_6 0x1
- #define RFCOMM_RPN_DATA_7 0x2
- #define RFCOMM_RPN_DATA_8 0x3
- #define RFCOMM_RPN_STOP_1 0
- #define RFCOMM_RPN_STOP_15 1
- #define RFCOMM_RPN_PARITY_NONE 0x0
- #define RFCOMM_RPN_PARITY_ODD 0x4
- #define RFCOMM_RPN_PARITY_EVEN 0x5
- #define RFCOMM_RPN_PARITY_MARK 0x6
- #define RFCOMM_RPN_PARITY_SPACE 0x7
- #define RFCOMM_RPN_FLOW_NONE 0x00
- #define RFCOMM_RPN_XON_CHAR 0x11
- #define RFCOMM_RPN_XOFF_CHAR 0x13
- #define RFCOMM_RPN_PM_BITRATE 0x0001
- #define RFCOMM_RPN_PM_DATA 0x0002
- #define RFCOMM_RPN_PM_STOP 0x0004
- #define RFCOMM_RPN_PM_PARITY 0x0008
- #define RFCOMM_RPN_PM_PARITY_TYPE 0x0010
- #define RFCOMM_RPN_PM_XON 0x0020
- #define RFCOMM_RPN_PM_XOFF 0x0040
- #define RFCOMM_RPN_PM_FLOW 0x3F00
- #define RFCOMM_RPN_PM_ALL 0x3F7F
- #define RFCOMM_DLICI(b) (((b) & 0xfc) >> 2)
- #define RFCOMM_CHANNEL(b) (((b) & 0xf8) >> 3)
- #define RFCOMM_DIRECTION(b) (((b) & 0x04) >> 2)
- #define RFCOMM_TYPE(b) (((b) & 0xef))
- #define RFCOMM_EA(b) (((b) & 0x01))
- #define RFCOMM_CR(b) (((b) & 0x02) >> 1)
- #define RFCOMM_PF(b) (((b) & 0x10) >> 4)
- #define RFCOMM_SRVCHANNEL(dlci) ((dlci) >> 1)
- #define RFCOMM_MKADDRESS(cr, dlci) (((dlci) & 0x3f) << 2) | ((cr) << 1) | 0x01)
- #define RFCOMM_MKCONTROL(type, pf) (((type) & 0xef) | ((pf) << 4))
- #define RFCOMM_MKDLICI(dir, channel) (((channel) & 0x1f) << 1) | (dir))
- #define RFCOMM_MKLEN8(len) (((len) << 1) | 1)
- #define RFCOMM_MKLEN16(len) ((len) << 1)
- #define RFCOMM_MCC_TYPE(b) (((b) & 0xfc) >> 2)
- #define RFCOMM_MCC_LENGTH(b) (((b) & 0xfe) >> 1)
- #define RFCOMM_MKMCC_TYPE(cr, type) (((type) << 2) | ((cr) << 1) | 0x01))
- #define RFCOMM_RPN_DATA_BITS(line) ((line) & 0x3)
- #define RFCOMM_RPN_STOP_BITS(line) (((line) >> 2) & 0x1)
- #define RFCOMM_RPN_PARITY(line) (((line) >> 3) & 0x3)
- #define RFCOMM_MKRPN_LINE_SETTINGS(data, stop, parity) (((data) & 0x3) | (((stop) & 0x1) << 2) | (((parity) & 0x3) << 3))

- #define `NG_BT_SOCKET_RFCOMM_SENDSPACE` (RFCOMM_MAX_CREDITS * RFCOMM_DEFAULT_MTU * 2)
- #define `NG_BT_SOCKET_RFCOMM_RECVSPACE` (RFCOMM_MAX_CREDITS * RFCOMM_DEFAULT_MTU * 2)
- #define `NG_BT_SOCKET_RFCOMM_SESSION_CLOSED` 0
- #define `NG_BT_SOCKET_RFCOMM_SESSION_LISTENING` 1
- #define `NG_BT_SOCKET_RFCOMM_SESSION_CONNECTING` 2
- #define `NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED` 3
- #define `NG_BT_SOCKET_RFCOMM_SESSION_OPEN` 4
- #define `NG_BT_SOCKET_RFCOMM_SESSION_DISCONNECTING` 5
- #define `NG_BT_SOCKET_RFCOMM_SESSION_INITIATOR` (1 << 0)
- #define `NG_BT_SOCKET_RFCOMM_SESSION_LFC` (1 << 1)
- #define `NG_BT_SOCKET_RFCOMM_SESSION_RFC` (1 << 2)
- #define `INITIATOR(s)` (((s) → flags & NG_BT_SOCKET_RFCOMM_SESSION_INITIATOR)? 1 : 0)
- #define `NG_BT_SOCKET_RFCOMM_DLC_TIMO` (1 << 0)
- #define `NG_BT_SOCKET_RFCOMM_DLC_CFC` (1 << 1)
- #define `NG_BT_SOCKET_RFCOMM_DLC_TIMEDOUT` (1 << 2)
- #define `NG_BT_SOCKET_RFCOMM_DLC_DETACHED` (1 << 3)
- #define `NG_BT_SOCKET_RFCOMM_DLC_SENDING` (1 << 4)
- #define `NG_BT_SOCKET_RFCOMM_DLC_CLOSED` 0
- #define `NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT` 1
- #define `NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING` 2
- #define `NG_BT_SOCKET_RFCOMM_DLC_CONNECTING` 3
- #define `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED` 4
- #define `NG_BT_SOCKET_RFCOMM_DLC_DISCONNECTING` 5
- #define `so2rfcomm_pcb(so)` ((struct `ng_btsocket_rfcomm_pcb` *)((so) → so_pcb))

Typedefs

- typedef `ng_btsocket_rfcomm_session` `ng_btsocket_rfcomm_session_t`
- typedef `ng_btsocket_rfcomm_session` * `ng_btsocket_rfcomm_session_p`
- typedef `ng_btsocket_rfcomm_pcb` `ng_btsocket_rfcomm_pcb_t`
- typedef `ng_btsocket_rfcomm_pcb` * `ng_btsocket_rfcomm_pcb_p`

Functions

- void `ng_btsocket_rfcomm_init` (void)
- void `ng_btsocket_rfcomm_abort` (struct socket *)
- void `ng_btsocket_rfcomm_close` (struct socket *)
- int `ng_btsocket_rfcomm_accept` (struct socket *, struct sockaddr **)
- int `ng_btsocket_rfcomm_attach` (struct socket *, int, struct thread *)
- int `ng_btsocket_rfcomm_bind` (struct socket *, struct sockaddr *, struct thread *)
- int `ng_btsocket_rfcomm_connect` (struct socket *, struct sockaddr *, struct thread *)
- int `ng_btsocket_rfcomm_control` (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)
- int `ng_btsocket_rfcomm_ctloutput` (struct socket *, struct sockopt *)
- void `ng_btsocket_rfcomm_detach` (struct socket *)
- int `ng_btsocket_rfcomm_disconnect` (struct socket *)
- int `ng_btsocket_rfcomm_listen` (struct socket *, int, struct thread *)

- int `ng_btsocket_rfcomm_peeraddr` (struct socket *, struct sockaddr **)
- int `ng_btsocket_rfcomm_send` (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)
- int `ng_btsocket_rfcomm_sockaddr` (struct socket *, struct sockaddr **)

Variables

- `rfcomm_frame_hdr` packed
- `rfcomm_cmd_hdr` packed
- `rfcomm_mcc_hdr` packed
- `rfcomm_mcc_msc` packed
- `rfcomm_mcc_rpn` packed
- `rfcomm_mcc_rls` packed
- `rfcomm_mcc_pn` packed

7.47.1 Define Documentation

7.47.1.1 #define INITIATOR(s) (((s) → flags & NG_BT_SOCKET_RFCOMM_SESSION_INITIATOR)? 1 : 0)

Definition at line 248 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_mcc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, and `ng_btsocket_rfcomm_send_pn()`.

7.47.1.2 #define NG_BT_SOCKET_RFCOMM_DLC_CFC (1 << 1)

Definition at line 272 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_attach()`, `ng_btsocket_rfcomm_ctloutput()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_send_pn()`, and `ng_btsocket_rfcomm_set_pn()`.

7.47.1.3 #define NG_BT_SOCKET_RFCOMM_DLC_CLOSED 0

Definition at line 278 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_attach()`, `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_disconnect()`, and `ng_btsocket_rfcomm_pcb_kill()`.

7.47.1.4 #define NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING 2

Definition at line 280 of file `ng_btsocket_rfcomm.h`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_connect_cfm()`, `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_disconnect()`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_receive_pn()`, and `ng_btsocket_rfcomm_session_process_pcb()`.

7.47.1.5 #define NG_BT_SOCKET_RFCOMM_DLC_CONNECTED 4

Definition at line 282 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_dm(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_receive_uih(), ng_btsocket_rfcomm_send(), ng_btsocket_rfcomm_session_clean(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.6 #define NG_BT_SOCKET_RFCOMM_DLC_CONNECTING 3

Definition at line 281 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.7 #define NG_BT_SOCKET_RFCOMM_DLC_DETACHED (1 << 3)

Definition at line 274 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_detach(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.8 #define NG_BT_SOCKET_RFCOMM_DLC_DISCONNECTING 5

Definition at line 283 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_receive_ua(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.9 #define NG_BT_SOCKET_RFCOMM_DLC_SENDING (1 << 4)

Definition at line 275 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_send(), and ng_btsocket_rfcomm_send().

7.47.1.10 #define NG_BT_SOCKET_RFCOMM_DLC_TIMEOUT (1 << 2)

Definition at line 273 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_session_process_pcb(), ng_btsocket_rfcomm_timeout(), and ng_btsocket_rfcomm_untimeout().

7.47.1.11 #define NG_BT_SOCKET_RFCOMM_DLC_TIMO (1 << 0)

Definition at line 271 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_session_process_pcb(), ng_btsocket_rfcomm_timeout(), and ng_btsocket_rfcomm_untimeout().

7.47.1.12 #define NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT 1

Definition at line 279 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_connect_cfm(), ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_process_timeout(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.13 #define NG_BT_SOCKET_RFCOMM_RECVSPACE (RFCOMM_MAX_CREDITS * RFCOMM_DEFAULT_MTU * 2)

Definition at line 222 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach().

7.47.1.14 #define NG_BT_SOCKET_RFCOMM_SENDSIZE (RFCOMM_MAX_CREDITS * RFCOMM_DEFAULT_MTU * 2)

Definition at line 220 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach().

7.47.1.15 #define NG_BT_SOCKET_RFCOMM_SESSION_CLOSED 0

Definition at line 236 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_dm(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_session_create(), ng_btsocket_rfcomm_session_task(), and ng_btsocket_rfcomm_sessions_task().

7.47.1.16 #define NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED 3

Definition at line 239 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_session_accept(), ng_btsocket_rfcomm_session_connect(), and ng_btsocket_rfcomm_session_task().

7.47.1.17 #define NG_BT_SOCKET_RFCOMM_SESSION_CONNECTING 2

Definition at line 238 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_session_create(), and ng_btsocket_rfcomm_session_task().

7.47.1.18 #define NG_BT_SOCKET_RFCOMM_SESSION_DISCONNECTING 5

Definition at line 241 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_ua(), and ng_btsocket_rfcomm_session_task().

7.47.1.19 #define NG_BT_SOCKET_RFCOMM_SESSION_INITIATOR (1 << 0)

Definition at line 244 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_session_create().

7.47.1.20 #define NG_BT_SOCKET_RFCOMM_SESSION_LFC (1 << 1)

Definition at line 245 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_uih().

7.47.1.21 #define NG_BT_SOCKET_RFCOMM_SESSION_LISTENING 1

Definition at line 237 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_listen(), ng_btsocket_rfcomm_session_create(), and ng_btsocket_rfcomm_session_task().

7.47.1.22 #define NG_BT_SOCKET_RFCOMM_SESSION_OPEN 4

Definition at line 240 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), and ng_btsocket_rfcomm_session_task().

7.47.1.23 #define NG_BT_SOCKET_RFCOMM_SESSION_RFC (1 << 2)

Definition at line 246 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc().

7.47.1.24 #define RFCOMM_CHANNEL(b) (((b) & 0xf8) >> 3)

Definition at line 183 of file ng_btsocket_rfcomm.h.

7.47.1.25 #define RFCOMM_CR(b) (((b) & 0x02) >> 1)

Definition at line 188 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), and ng_btsocket_rfcomm_receive_test().

7.47.1.26 #define RFCOMM_DEFAULT_CREDITS 7

Definition at line 48 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach(), and ng_btsocket_rfcomm_receive_pn().

7.47.1.27 #define RFCOMM_DEFAULT_MTU 667

Definition at line 45 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach(), and ng_btsocket_rfcomm_session_create().

7.47.1.28 #define RFCOMM_DIRECTION(b) (((b) & 0x04) >> 2)

Definition at line 184 of file ng_btsocket_rfcomm.h.

7.47.1.29 #define RFCOMM_DLCI(b) (((b) & 0xfc) >> 2)

Definition at line 182 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_rls(), and ng_btsocket_rfcomm_receive_rpn().

7.47.1.30 #define RFCOMM_EA(b) (((b) & 0x01))

Definition at line 187 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_frame().

7.47.1.31 #define RFCOMM_FRAME_DISC 0x43

Definition at line 53 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_send_command(), and ng_btsocket_rfcomm_session_process_pcb().

7.47.1.32 #define RFCOMM_FRAME_DM 0x0f

Definition at line 55 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_receive_uih(), and ng_btsocket_rfcomm_send_command().

7.47.1.33 #define RFCOMM_FRAME_SABM 0x2f

Definition at line 52 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_send_command(), and ng_btsocket_rfcomm_session_connect().

7.47.1.34 #define RFCOMM_FRAME_UA 0x63

Definition at line 54 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_sabm(), and ng_btsocket_rfcomm_send_command().

7.47.1.35 #define RFCOMM_FRAME_UIH 0xef

Definition at line 56 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_check_fcs(), and ng_btsocket_rfcomm_send_uih().

7.47.1.36 #define RFCOMM_MAX_CREDITS 40

Definition at line 49 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_receive_uih(), and ng_btsocket_rfcomm_send_credits().

7.47.1.37 #define RFCOMM_MAX_MTU 1024

Definition at line 46 of file ng_btsocket_rfcomm.h.

7.47.1.38 #define RFCOMM_MCC_FCOFF 0x18

Definition at line 61 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc().

7.47.1.39 #define RFCOMM_MCC_FCON 0x28

Definition at line 60 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc(), and ng_btsocket_rfcomm_receive_mcc().

7.47.1.40 #define RFCOMM_MCC_LENGTH(b) (((b) & 0xfe) >> 1)

Definition at line 204 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), and ng_btsocket_rfcomm_receive_test().

7.47.1.41 #define RFCOMM_MCC_MSC 0x38

Definition at line 62 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), and ng_btsocket_rfcomm_send_msc().

7.47.1.42 #define RFCOMM_MCC_NSC 0x04

Definition at line 66 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc().

7.47.1.43 #define RFCOMM_MCC_PN 0x20

Definition at line 65 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_pn(), and ng_btsocket_rfcomm_send_pn().

7.47.1.44 #define RFCOMM_MCC_RLS 0x14

Definition at line 64 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), and ng_btsocket_rfcomm_receive_rls().

7.47.1.45 #define RFCOMM_MCC_RPN 0x24

Definition at line 63 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), and ng_btsocket_rfcomm_receive_rpn().

7.47.1.46 #define RFCOMM_MCC_TEST 0x08

Definition at line 59 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), and ng_btsocket_rfcomm_receive_test().

7.47.1.47 #define RFCOMM_MCC_TYPE(b) (((b) & 0xfc) >> 2)

Definition at line 203 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc(), and ng_btsocket_rfcomm_receive_mcc().

7.47.1.48 #define RFCOMM_MKADDRESS(cr, dlci) (((dlci) & 0x3f) << 2) | ((cr) << 1) | 0x01

Definition at line 193 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_credits(), ng_btsocket_rfcomm_send_msc(), and ng_btsocket_rfcomm_send_pn().

7.47.1.49 #define RFCOMM_MKCONTROL(type, pf) (((type) & 0xef) | ((pf) << 4))

Definition at line 196 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_send_command(), and ng_btsocket_rfcomm_send_uih().

7.47.1.50 #define RFCOMM_MKDLCI(dir, channel) (((channel) & 0x1f) << 1) | (dir)

Definition at line 197 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_connect().

7.47.1.51 #define RFCOMM_MKLEN16(len) ((len) << 1)

Definition at line 200 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_send_uih().

7.47.1.52 #define RFCOMM_MKLEN8(len) (((len) << 1) | 1)

Definition at line 199 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_msc(), ng_btsocket_rfcomm_send_pn(), and ng_btsocket_rfcomm_send_uih().

7.47.1.53 #define RFCOMM_MKMCC_TYPE(cr, type) (((type) << 2) | ((cr) << 1) | 0x01)

Definition at line 205 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_send_msc(), and ng_btsocket_rfcomm_send_pn().

7.47.1.54 #define RFCOMM_MKRPN_LINE_SETTINGS(data, stop, parity) (((data) & 0x3) | (((stop) & 0x1) << 2) | (((parity) & 0x3) << 3))

Definition at line 211 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.55 #define RFCOMM_MODEM_DV 0x80

Definition at line 73 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach().

7.47.1.56 #define RFCOMM_MODEM_FC 0x02

Definition at line 69 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_receive_msc(), and ng_btsocket_rfcomm_receive_uih().

7.47.1.57 #define RFCOMM_MODEM_IC 0x40

Definition at line 72 of file ng_btsocket_rfcomm.h.

7.47.1.58 #define RFCOMM_MODEM_RTC 0x04

Definition at line 70 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach().

7.47.1.59 #define RFCOMM_MODEM_RTR 0x08

Definition at line 71 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach().

7.47.1.60 #define RFCOMM_PF(b) (((b) & 0x10) >> 4)

Definition at line 189 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_frame().

7.47.1.61 #define RFCOMM_RPN_BR_115200 0x7

Definition at line 83 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.62 #define RFCOMM_RPN_BR_19200 0x4

Definition at line 80 of file ng_btsocket_rfcomm.h.

7.47.1.63 #define RFCOMM_RPN_BR_230400 0x8

Definition at line 84 of file ng_btsocket_rfcomm.h.

7.47.1.64 #define RFCOMM_RPN_BR_2400 0x0

Definition at line 76 of file ng_btsocket_rfcomm.h.

7.47.1.65 #define RFCOMM_RPN_BR_38400 0x5

Definition at line 81 of file ng_btsocket_rfcomm.h.

7.47.1.66 #define RFCOMM_RPN_BR_4800 0x1

Definition at line 77 of file ng_btsocket_rfcomm.h.

7.47.1.67 #define RFCOMM_RPN_BR_57600 0x6

Definition at line 82 of file ng_btsocket_rfcomm.h.

7.47.1.68 #define RFCOMM_RPN_BR_7200 0x2

Definition at line 78 of file ng_btsocket_rfcomm.h.

7.47.1.69 #define RFCOMM_RPN_BR_9600 0x3

Definition at line 79 of file ng_btsocket_rfcomm.h.

7.47.1.70 #define RFCOMM_RPN_DATA_5 0x0

Definition at line 87 of file ng_btsocket_rfcomm.h.

7.47.1.71 #define RFCOMM_RPN_DATA_6 0x1

Definition at line 88 of file ng_btsocket_rfcomm.h.

7.47.1.72 #define RFCOMM_RPN_DATA_7 0x2

Definition at line 89 of file ng_btsocket_rfcomm.h.

7.47.1.73 #define RFCOMM_RPN_DATA_8 0x3

Definition at line 90 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.74 #define RFCOMM_RPN_DATA_BITS(line) ((line) & 0x3)

Definition at line 208 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.75 #define RFCOMM_RPN_FLOW_NONE 0x00

Definition at line 104 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.76 #define RFCOMM_RPN_PARITY(line) (((line) >> 3) & 0x3)

Definition at line 210 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.77 #define RFCOMM_RPN_PARITY_EVEN 0x5

Definition at line 99 of file ng_btsocket_rfcomm.h.

7.47.1.78 #define RFCOMM_RPN_PARITY_MARK 0x6

Definition at line 100 of file ng_btsocket_rfcomm.h.

7.47.1.79 #define RFCOMM_RPN_PARITY_NONE 0x0

Definition at line 97 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.80 #define RFCOMM_RPN_PARITY_ODD 0x4

Definition at line 98 of file ng_btsocket_rfcomm.h.

7.47.1.81 #define RFCOMM_RPN_PARITY_SPACE 0x7

Definition at line 101 of file ng_btsocket_rfcomm.h.

7.47.1.82 #define RFCOMM_RPN_PM_ALL 0x3F7F

Definition at line 117 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.83 #define RFCOMM_RPN_PM_BITRATE 0x0001

Definition at line 109 of file ng_btsocket_rfcomm.h.

7.47.1.84 #define RFCOMM_RPN_PM_DATA 0x0002

Definition at line 110 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.85 #define RFCOMM_RPN_PM_FLOW 0x3F00

Definition at line 116 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.86 #define RFCOMM_RPN_PM_PARITY 0x0008

Definition at line 112 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.87 #define RFCOMM_RPN_PM_PARITY_TYPE 0x0010

Definition at line 113 of file ng_btsocket_rfcomm.h.

7.47.1.88 #define RFCOMM_RPN_PM_STOP 0x0004

Definition at line 111 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.89 #define RFCOMM_RPN_PM_XOFF 0x0040

Definition at line 115 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.90 #define RFCOMM_RPN_PM_XON 0x0020

Definition at line 114 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.91 #define RFCOMM_RPN_STOP_1 0

Definition at line 93 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.92 #define RFCOMM_RPN_STOP_15 1

Definition at line 94 of file ng_btsocket_rfcomm.h.

7.47.1.93 #define RFCOMM_RPN_STOP_BITS(line) (((line) >> 2) & 0x1)

Definition at line 209 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.94 #define RFCOMM_RPN_XOFF_CHAR 0x13

Definition at line 106 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.95 #define RFCOMM_RPN_XON_CHAR 0x11

Definition at line 105 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_rpn().

7.47.1.96 #define RFCOMM_SRVCHANNEL(dlci) ((dlci) >> 1)

Definition at line 191 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_pn(), and ng_btsocket_rfcomm_receive_sabm().

7.47.1.97 #define RFCOMM_TYPE(b) (((b) & 0xef))

Definition at line 185 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_receive_frame().

7.47.1.98 #define so2rfcomm_pcb(so) ((struct ng_btsocket_rfcomm_pcb *)((so) → so_pcb))

Definition at line 307 of file ng_btsocket_rfcomm.h.

Referenced by ng_btsocket_rfcomm_attach(), ng_btsocket_rfcomm_bind(), ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_connect_ind(), ng_btsocket_rfcomm_ctloutput(), ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_listen(), ng_btsocket_rfcomm_peeraddr(), ng_btsocket_rfcomm_send(), and ng_btsocket_rfcomm_sockaddr().

7.47.2 Typedef Documentation

7.47.2.1 typedef struct ng_btsocket_rfcomm_pcb* ng_btsocket_rfcomm_pcb_p

Definition at line 305 of file ng_btsocket_rfcomm.h.

7.47.2.2 typedef struct ng_btsocket_rfcomm_pcb ng_btsocket_rfcomm_pcb_t

Definition at line 304 of file ng_btsocket_rfcomm.h.

7.47.2.3 typedef struct ng_btsocket_rfcomm_session* ng_btsocket_rfcomm_session_p

Definition at line 260 of file ng_btsocket_rfcomm.h.

7.47.2.4 typedef struct ng_btsocket_rfcomm_session ng_btsocket_rfcomm_session_t

Definition at line 259 of file ng_btsocket_rfcomm.h.

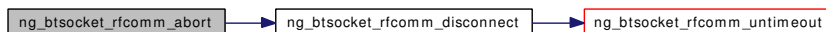
7.47.3 Function Documentation

7.47.3.1 void ng_btsocket_rfcomm_abort (struct socket *)

Definition at line 347 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_disconnect().

Here is the call graph for this function:



7.47.3.2 int ng_btsocket_rfcomm_accept (struct socket *, struct sockaddr **)

Definition at line 367 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_peeraddr().

Here is the call graph for this function:



7.47.3.3 int ng_btsocket_rfcomm_attach (struct socket *, int, struct thread *)

Definition at line 377 of file ng_btsocket_rfcomm.c.

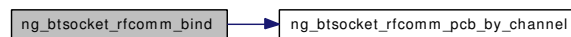
References BLUETOOTH_PROTO_RFCOMM, M_NETGRAPH_BTSTACK_RFCOMM, NG_BTSTACK_RFCOMM_DLC_CFC, NG_BTSTACK_RFCOMM_DLC_CLOSED, NG_BTSTACK_RFCOMM_RECVSPACE, NG_BTSTACK_RFCOMM_SENDSAPCE, RFCOMM_DEFAULT_CREDITS, RFCOMM_DEFAULT_MTU, RFCOMM_MODEM_DV, RFCOMM_MODEM_RTC, RFCOMM_MODEM_RTR, and so2rfcomm_pcb.

7.47.3.4 int ng_btsocket_rfcomm_bind (struct socket *, struct sockaddr *, struct thread *)

Definition at line 440 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::channel, ng_btsocket_rfcomm_pcb_by_channel(), sockaddr_rfcomm::rfcomm_family, sockaddr_rfcomm::rfcomm_len, so2rfcomm_pcb, and ng_btsocket_rfcomm_pcb::src.

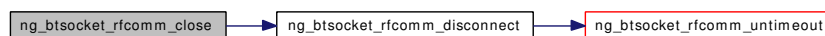
Here is the call graph for this function:

**7.47.3.5 void ng_btsocket_rfcomm_close (struct socket *)**

Definition at line 355 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_disconnect().

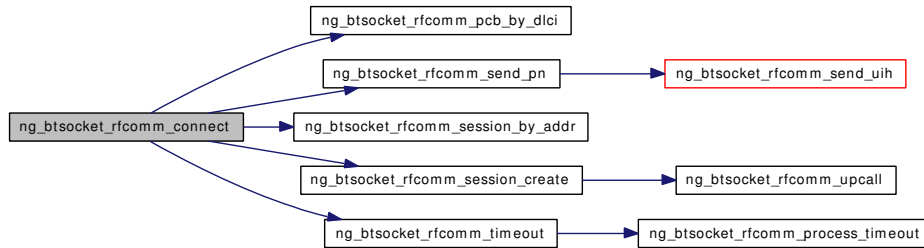
Here is the call graph for this function:

**7.47.3.6 int ng_btsocket_rfcomm_connect (struct socket *, struct sockaddr *, struct thread *)**

Definition at line 473 of file ng_btsocket_rfcomm.c.

References BLUETOOTH_PROTO_L2CAP, ng_btsocket_rfcomm_pcb::channel, ng_btsocket_rfcomm_pcb::dlci, ng_btsocket_rfcomm_pcb::dst, INITIATOR, ng_btsocket_rfcomm_session::l2so, ng_btsocket_rfcomm_pcb::mtu, ng_btsocket_rfcomm_session::mtu, NG_BTSTACK_RFCOMM_DLC_CONFIGURING, NG_BTSTACK_RFCOMM_DLC_W4_CONNECT, ng_btsocket_rfcomm_pcb_by_dlci(), ng_btsocket_rfcomm_send_pn(), ng_btsocket_rfcomm_session_by_addr(), NG_BTSTACK_RFCOMM_SESSION_CONNECTED, NG_BTSTACK_RFCOMM_SESSION_CONNECTING, ng_btsocket_rfcomm_session_create(), NG_BTSTACK_RFCOMM_SESSION_OPEN, ng_btsocket_rfcomm_task_wakeup, ng_btsocket_rfcomm_timeout(), NG_HCI_BDADDR_ANY, ng_btsocket_rfcomm_pcb::pcb_mtx, sockaddr_rfcomm::rfcomm_family, sockaddr_rfcomm::rfcomm_len, RFCOMM_MKDLCI, ng_btsocket_rfcomm_pcb::session, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_pcb::so, so2l2cap_pcb, so2rfcomm_pcb, ng_btsocket_rfcomm_pcb::src, ng_btsocket_rfcomm_session::state, and ng_btsocket_rfcomm_pcb::state.

Here is the call graph for this function:



7.47.3.7 int ng_btsocket_rfcomm_control (struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)

Definition at line 613 of file ng_btsocket_rfcomm.c.

7.47.3.8 int ng_btsocket_rfcomm_ctloutput (struct socket *, struct sockopt *)

Definition at line 624 of file ng_btsocket_rfcomm.c.

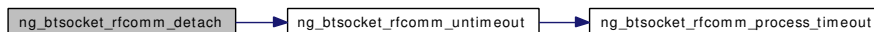
References ng_btsocket_rfcomm_fc_info::cfc, ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_fc_info::lmodem, ng_btsocket_rfcomm_pcb::lmodem, ng_btsocket_rfcomm_pcb::mtu, NG_BTSOCKET_RFCOMM_DLC_CFC, ng_btsocket_rfcomm_pcb::pcb_mtx, ng_btsocket_rfcomm_fc_info::reserved, ng_btsocket_rfcomm_fc_info::rmodem, ng_btsocket_rfcomm_pcb::rmodem, ng_btsocket_rfcomm_fc_info::rx_cred, ng_btsocket_rfcomm_pcb::rx_cred, so2rfcomm_pcb, SO_RFCOMM_FC_INFO, SO_RFCOMM_MTU, SOL_RFCOMM, ng_btsocket_rfcomm_fc_info::tx_cred, and ng_btsocket_rfcomm_pcb::tx_cred.

7.47.3.9 void ng_btsocket_rfcomm_detach (struct socket *)

Definition at line 685 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::flags, M_NETGRAPH_BTSOCKET_RFCOMM, NG_BTSOCKET_RFCOMM_DLC_CLOSED, NG_BTSOCKET_RFCOMM_DLC_CONFIGURING, NG_BTSOCKET_RFCOMM_DLC_CONNECTED, NG_BTSOCKET_RFCOMM_DLC_CONNECTING, NG_BTSOCKET_RFCOMM_DLC_DETACHED, NG_BTSOCKET_RFCOMM_DLC_DISCONNECTING, NG_BTSOCKET_RFCOMM_DLC_TIMO, NG_BTSOCKET_RFCOMM_DLC_W4_CONNECT, ng_btsocket_rfcomm_task_wakeup, ng_btsocket_rfcomm_untimeout(), ng_btsocket_rfcomm_pcb::pcb_mtx, ng_btsocket_rfcomm_pcb::session, so2rfcomm_pcb, and ng_btsocket_rfcomm_pcb::state.

Here is the call graph for this function:



7.47.3.10 int ng_btsocket_rfcomm_disconnect (struct socket *)

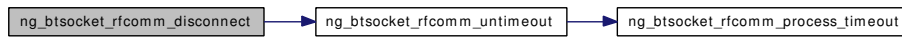
Definition at line 743 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::flags, NG_BTSOCKET_RFCOMM_DLC_CLOSED, NG_BTSOCKET_RFCOMM_DLC_CONFIGURING, NG_BTSOCKET_RFCOMM_DLC_CONNECTED,

NG_BT_SOCKET_RFCOMM_DLC_CONNECTING, NG_BT_SOCKET_RFCOMM_DLC_DISCONNECTING, NG_BT_SOCKET_RFCOMM_DLC_TIMO, NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT, ng_btsocket_rfcomm_task_wakeup, ng_btsocket_rfcomm_untimeout(), ng_btsocket_rfcomm_pcb::pcb_mtx, so2rfcomm_pcb, and ng_btsocket_rfcomm_pcb::state.

Referenced by ng_btsocket_rfcomm_abort(), and ng_btsocket_rfcomm_close().

Here is the call graph for this function:

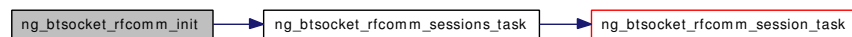


7.47.3.11 void ng_btsocket_rfcomm_init (void)

Definition at line 322 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_debug_level, ng_btsocket_rfcomm_sessions_task(), ng_btsocket_rfcomm_task, ng_btsocket_rfcomm_timo, and NG_BT_SOCKET_WARN_LEVEL.

Here is the call graph for this function:

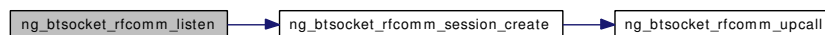


7.47.3.12 int ng_btsocket_rfcomm_listen (struct socket *, int, struct thread *)

Definition at line 797 of file ng_btsocket_rfcomm.c.

References BLUETOOTH_PROTO_L2CAP, ng_btsocket_rfcomm_pcb::channel, ng_btsocket_rfcomm_session_create(), NG_BT_SOCKET_RFCOMM_SESSION_LISTENING, NG_HCI_BDADDR_ANY, so2rfcomm_pcb, and ng_btsocket_rfcomm_session::state.

Here is the call graph for this function:



7.47.3.13 int ng_btsocket_rfcomm_peeraddr (struct socket *, struct sockaddr **)

Definition at line 881 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::channel, ng_btsocket_rfcomm_pcb::dst, sockaddr_rfcomm::rfcomm_bdaddr, sockaddr_rfcomm::rfcomm_channel, sockaddr_rfcomm::rfcomm_family, sockaddr_rfcomm::rfcomm_len, and so2rfcomm_pcb.

Referenced by ng_btsocket_rfcomm_accept().

7.47.3.14 int ng_btsocket_rfcomm_send (struct socket *, int, struct mbuf *, struct sockaddr *, struct mbuf *, struct thread *)

Definition at line 904 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_pcb::flags`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED`, `NG_BT_SOCKET_RFCOMM_DLC_SENDING`, `ng_btsocket_rfcomm_task_wakeup`, `NG_FREE_M`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_pcb::so`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::state`.

7.47.3.15 `int ng_btsocket_rfcomm_sockaddr (struct socket *, struct sockaddr **)`

Definition at line 947 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::channel`, `sockaddr_rfcomm::rfcomm_bdaddr`, `sockaddr_rfcomm::rfcomm_channel`, `sockaddr_rfcomm::rfcomm_family`, `sockaddr_rfcomm::rfcomm_len`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::src`.

7.47.4 Variable Documentation

7.47.4.1 `struct rfcomm_mcc_pn` packed

7.47.4.2 `struct rfcomm_mcc_rls` packed

7.47.4.3 `struct rfcomm_mcc_rpn` packed

7.47.4.4 `struct rfcomm_mcc_msc` packed

7.47.4.5 `struct rfcomm_mcc_hdr` packed

7.47.4.6 `struct rfcomm_cmd_hdr` packed

7.47.4.7 `struct rfcomm_frame_hdr` packed

Referenced by `con_compl()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_qos_req()`, `ng_hci_lp_sco_con_req()`, and `ng_l2cap_l2ca_clt_receive()`.

7.48 /usr/src/sys/netgraph/bluetooth/include/ng_h4.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_h4_node_stat_ep](#)

Defines

- #define [NG_H4_HOOK](#) "hook"
- #define [NG_H4_NODE_TYPE](#) "h4"
- #define [NGM_H4_COOKIE](#) 1013899512
- #define [NG_H4_W4_PKT_IND](#) 1
- #define [NG_H4_W4_PKT_HDR](#) 2
- #define [NG_H4_W4_PKT_DATA](#) 3
- #define [NG_H4_ALERT_LEVEL](#) 1
- #define [NG_H4_ERR_LEVEL](#) 2
- #define [NG_H4_WARN_LEVEL](#) 3
- #define [NG_H4_INFO_LEVEL](#) 4
- #define [NGM_H4_NODE_RESET](#) 1
- #define [NGM_H4_NODE_GET_STATE](#) 2
- #define [NGM_H4_NODE_GET_DEBUG](#) 3
- #define [NGM_H4_NODE_SET_DEBUG](#) 4
- #define [NGM_H4_NODE_GET_QLEN](#) 5
- #define [NGM_H4_NODE_SET_QLEN](#) 6
- #define [NGM_H4_NODE_GET_STAT](#) 7
- #define [NGM_H4_NODE_RESET_STAT](#) 8

Typedefs

- typedef u_int16_t [ng_h4_node_state_ep](#)
- typedef u_int16_t [ng_h4_node_debug_ep](#)
- typedef int32_t [ng_h4_node_qlen_ep](#)

7.48.1 Define Documentation

7.48.1.1 #define [NG_H4_ALERT_LEVEL](#) 1

Definition at line 70 of file [ng_h4.h](#).

7.48.1.2 #define [NG_H4_ERR_LEVEL](#) 2

Definition at line 71 of file [ng_h4.h](#).

7.48.1.3 #define NG_H4_HOOK "hook"

Definition at line 58 of file ng_h4.h.

Referenced by ng_h4_newhook(), and ng_h4_rcvmsg().

7.48.1.4 #define NG_H4_INFO_LEVEL 4

Definition at line 73 of file ng_h4.h.

7.48.1.5 #define NG_H4_NODE_TYPE "h4"

Definition at line 61 of file ng_h4.h.

7.48.1.6 #define NG_H4_W4_PKT_DATA 3

Definition at line 67 of file ng_h4.h.

Referenced by ng_h4_input().

7.48.1.7 #define NG_H4_W4_PKT_HDR 2

Definition at line 66 of file ng_h4.h.

Referenced by ng_h4_input().

7.48.1.8 #define NG_H4_W4_PKT_IND 1

Definition at line 65 of file ng_h4.h.

Referenced by ng_h4_disconnect(), ng_h4_input(), ng_h4_open(), and ng_h4_rcvmsg().

7.48.1.9 #define NG_H4_WARN_LEVEL 3

Definition at line 72 of file ng_h4.h.

Referenced by ng_h4_open().

7.48.1.10 #define NGM_H4_COOKIE 1013899512

Definition at line 62 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.11 #define NGM_H4_NODE_GET_DEBUG 3

Definition at line 89 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.12 #define NGM_H4_NODE_GET_QLEN 5

Definition at line 94 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.13 #define NGM_H4_NODE_GET_STAT 7

Definition at line 99 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.14 #define NGM_H4_NODE_GET_STATE 2

Definition at line 85 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.15 #define NGM_H4_NODE_RESET 1

Definition at line 82 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.16 #define NGM_H4_NODE_RESET_STAT 8

Definition at line 110 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.17 #define NGM_H4_NODE_SET_DEBUG 4

Definition at line 90 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.1.18 #define NGM_H4_NODE_SET_QLEN 6

Definition at line 95 of file ng_h4.h.

Referenced by ng_h4_rcvmsg().

7.48.2 Typedef Documentation**7.48.2.1 typedef u_int16_t ng_h4_node_debug_ep**

Definition at line 91 of file ng_h4.h.

7.48.2.2 typedef int32_t ng_h4_node_qlen_ep

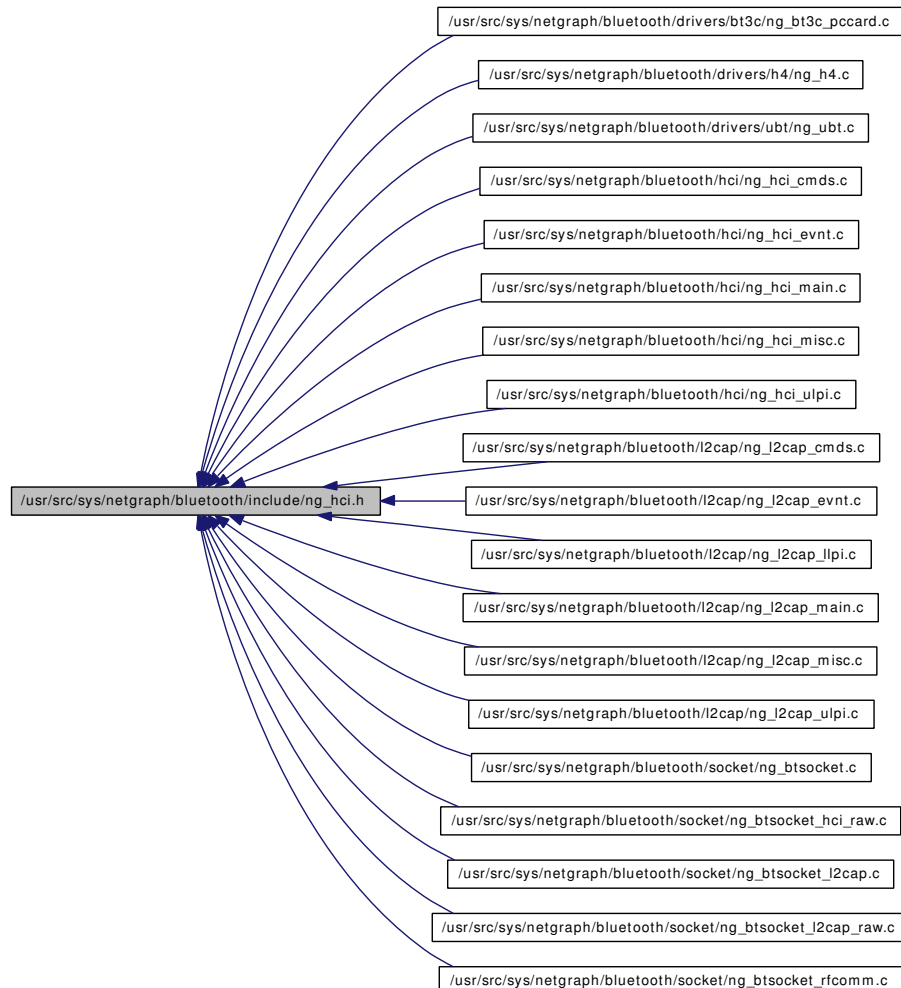
Definition at line 96 of file ng_h4.h.

7.48.2.3 `typedef u_int16_t ng_h4_node_state_ep`

Definition at line 86 of file `ng_h4.h`.

7.49 /usr/src/sys/netgraph/bluetooth/include/ng_hci.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [ng_hci_lp_con_req_ep](#)
- struct [ng_hci_lp_discon_req_ep](#)
- struct [ng_hci_lp_con_cfm_ep](#)
- struct [ng_hci_lp_con_ind_ep](#)
- struct [ng_hci_lp_con_rsp_ep](#)

- struct `__attribute__`
- struct `__attribute__`
- struct `__attribute__`
- struct `__attribute__`
- struct `__attribute__`

Defines

- #define `NG_HCI_NODE_TYPE` "hci"
- #define `NGM_HCI_COOKIE` 1000774184
- #define `NG_HCI_HOOK_DRV` "drv"
- #define `NG_HCI_HOOK_ACL` "acl"
- #define `NG_HCI_HOOK_SCO` "sco"
- #define `NG_HCI_HOOK_RAW` "raw"
- #define `NG_HCI_BDADDR_SIZE` 6
- #define `NG_HCI_LAP_SIZE` 3
- #define `NG_HCI_KEY_SIZE` 16
- #define `NG_HCI_PIN_SIZE` 16
- #define `NG_HCI_EVENT_MASK_SIZE` 8
- #define `NG_HCI_CLASS_SIZE` 3
- #define `NG_HCI_FEATURES_SIZE` 8
- #define `NG_HCI_UNIT_NAME_SIZE` 248
- #define `NG_HCI_SPEC_V10` 0x00
- #define `NG_HCI_SPEC_V11` 0x01
- #define `NG_HCI_LMP_3SLOT` 0x01
- #define `NG_HCI_LMP_5SLOT` 0x02
- #define `NG_HCI_LMP_ENCRYPTION` 0x04
- #define `NG_HCI_LMP_SLOT_OFFSET` 0x08
- #define `NG_HCI_LMP_TIMING_ACCURACY` 0x10
- #define `NG_HCI_LMP_SWITCH` 0x20
- #define `NG_HCI_LMP_HOLD_MODE` 0x40
- #define `NG_HCI_LMP_SNIFF_MODE` 0x80
- #define `NG_HCI_LMP_PARK_MODE` 0x01
- #define `NG_HCI_LMP_RSSI` 0x02
- #define `NG_HCI_LMP_CHANNEL_QUALITY` 0x04
- #define `NG_HCI_LMP_SCO_LINK` 0x08
- #define `NG_HCI_LMP_HV2_PKT` 0x10
- #define `NG_HCI_LMP_HV3_PKT` 0x20
- #define `NG_HCI_LMP_ULAW_LOG` 0x40
- #define `NG_HCI_LMP_ALAW_LOG` 0x80
- #define `NG_HCI_LMP_CVSD` 0x01
- #define `NG_HCI_LMP_PAGING_SCHEME` 0x02
- #define `NG_HCI_LMP_POWER_CONTROL` 0x04
- #define `NG_HCI_LMP_TRANSPARENT_SCO` 0x08
- #define `NG_HCI_LMP_FLOW_CONTROL_LAG0` 0x10
- #define `NG_HCI_LMP_FLOW_CONTROL_LAG1` 0x20
- #define `NG_HCI_LMP_FLOW_CONTROL_LAG2` 0x40
- #define `NG_HCI_LINK_SCO` 0x00
- #define `NG_HCI_LINK_ACL` 0x01
- #define `NG_HCI_PKT_DM1` 0x0008

- #define NG_HCI_PKT_DH1 0x0010
- #define NG_HCI_PKT_HV1 0x0020
- #define NG_HCI_PKT_HV2 0x0040
- #define NG_HCI_PKT_HV3 0x0080
- #define NG_HCI_PKT_DM3 0x0400
- #define NG_HCI_PKT_DH3 0x0800
- #define NG_HCI_PKT_DM5 0x4000
- #define NG_HCI_PKT_DH5 0x8000
- #define NG_HCI_UNIT_MODE_ACTIVE 0x00
- #define NG_HCI_UNIT_MODE_HOLD 0x01
- #define NG_HCI_UNIT_MODE_SNIFF 0x02
- #define NG_HCI_UNIT_MODE_PARK 0x03
- #define NG_HCI_MANDATORY_PAGE_SCAN_MODE 0x00
- #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE1 0x01
- #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE2 0x02
- #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE3 0x03
- #define NG_HCI_SCAN_REP_MODE0 0x00
- #define NG_HCI_SCAN_REP_MODE1 0x01
- #define NG_HCI_SCAN_REP_MODE2 0x02
- #define NG_HCI_PAGE_SCAN_PERIOD_MODE0 0x00
- #define NG_HCI_PAGE_SCAN_PERIOD_MODE1 0x01
- #define NG_HCI_PAGE_SCAN_PERIOD_MODE2 0x02
- #define NG_HCI_NO_SCAN_ENABLE 0x00
- #define NG_HCI_INQUIRY_ENABLE_PAGE_DISABLE 0x01
- #define NG_HCI_INQUIRY_DISABLE_PAGE_ENABLE 0x02
- #define NG_HCI_INQUIRY_ENABLE_PAGE_ENABLE 0x03
- #define NG_HCI_HOLD_MODE_NO_CHANGE 0x00
- #define NG_HCI_HOLD_MODE_SUSPEND_PAGE_SCAN 0x01
- #define NG_HCI_HOLD_MODE_SUSPEND_INQUIRY_SCAN 0x02
- #define NG_HCI_HOLD_MODE_SUSPEND_PERIOD_INQUIRY 0x04
- #define NG_HCI_ROLE_MASTER 0x00
- #define NG_HCI_ROLE_SLAVE 0x01
- #define NG_HCI_USE_SEMI_PERMANENT_LINK_KEYS 0x00
- #define NG_HCI_USE_TEMPORARY_LINK_KEY 0x01
- #define NG_HCI_PIN_TYPE_VARIABLE 0x00
- #define NG_HCI_PIN_TYPE_FIXED 0x01
- #define NG_HCI_LINK_KEY_TYPE_COMBINATION_KEY 0x00
- #define NG_HCI_LINK_KEY_TYPE_LOCAL_UNIT_KEY 0x01
- #define NG_HCI_LINK_KEY_TYPE_REMOTE_UNIT_KEY 0x02
- #define NG_HCI_ENCRYPTION_MODE_NONE 0x00
- #define NG_HCI_ENCRYPTION_MODE_P2P 0x01
- #define NG_HCI_ENCRYPTION_MODE_ALL 0x02
- #define NG_HCI_SERVICE_TYPE_NO_TRAFFIC 0x00
- #define NG_HCI_SERVICE_TYPE_BEST_EFFORT 0x01
- #define NG_HCI_SERVICE_TYPE_GUARANTEED 0x02
- #define NG_HCI_LINK_POLICY_DISABLE_ALL_LM_MODES 0x0000
- #define NG_HCI_LINK_POLICY_ENABLE_ROLE_SWITCH 0x0001
- #define NG_HCI_LINK_POLICY_ENABLE_HOLD_MODE 0x0002
- #define NG_HCI_LINK_POLICY_ENABLE_SNIFF_MODE 0x0004
- #define NG_HCI_LINK_POLICY_ENABLE_PARK_MODE 0x0008

- #define NG_HCI_EVMSK_ALL 0x00000000ffffff
- #define NG_HCI_EVMSK_NONE 0x0000000000000000
- #define NG_HCI_EVMSK_INQUIRY_COMPL 0x0000000000000001
- #define NG_HCI_EVMSK_INQUIRY_RESULT 0x0000000000000002
- #define NG_HCI_EVMSK_CON_COMPL 0x0000000000000004
- #define NG_HCI_EVMSK_CON_REQ 0x0000000000000008
- #define NG_HCI_EVMSK_DISCON_COMPL 0x0000000000000010
- #define NG_HCI_EVMSK_AUTH_COMPL 0x0000000000000020
- #define NG_HCI_EVMSK_REMOTE_NAME_REQ_COMPL 0x0000000000000040
- #define NG_HCI_EVMSK_ENCRYPTION_CHANGE 0x0000000000000080
- #define NG_HCI_EVMSK_CHANGE_CON_LINK_KEY_COMPL 0x0000000000000100
- #define NG_HCI_EVMSK_MASTER_LINK_KEY_COMPL 0x0000000000000200
- #define NG_HCI_EVMSK_READ_REMOTE_FEATURES_COMPL 0x0000000000000400
- #define NG_HCI_EVMSK_READ_REMOTE_VER_INFO_COMPL 0x0000000000000800
- #define NG_HCI_EVMSK_QOS_SETUP_COMPL 0x0000000000001000
- #define NG_HCI_EVMSK_COMMAND_COMPL 0x0000000000002000
- #define NG_HCI_EVMSK_COMMAND_STATUS 0x0000000000004000
- #define NG_HCI_EVMSK_HARDWARE_ERROR 0x0000000000008000
- #define NG_HCI_EVMSK_FLUSH_OCCUR 0x0000000000010000
- #define NG_HCI_EVMSK_ROLE_CHANGE 0x0000000000020000
- #define NG_HCI_EVMSK_NUM_COMPL_PKTS 0x0000000000040000
- #define NG_HCI_EVMSK_MODE_CHANGE 0x0000000000080000
- #define NG_HCI_EVMSK_RETURN_LINK_KEYS 0x0000000000100000
- #define NG_HCI_EVMSK_PIN_CODE_REQ 0x0000000000200000
- #define NG_HCI_EVMSK_LINK_KEY_REQ 0x0000000000400000
- #define NG_HCI_EVMSK_LINK_KEY_NOTIFICATION 0x0000000000800000
- #define NG_HCI_EVMSK_LOOPBACK_COMMAND 0x0000000001000000
- #define NG_HCI_EVMSK_DATA_BUFFER_OVERFLOW 0x0000000002000000
- #define NG_HCI_EVMSK_MAX_SLOT_CHANGE 0x0000000004000000
- #define NG_HCI_EVMSK_READ_CLOCK_OFFSET_COMLETE 0x0000000008000000
- #define NG_HCI_EVMSK_CON_PKT_TYPE_CHANGED 0x0000000010000000
- #define NG_HCI_EVMSK_QOS_VIOLATION 0x0000000020000000
- #define NG_HCI_EVMSK_PAGE_SCAN_MODE_CHANGE 0x0000000040000000
- #define NG_HCI_EVMSK_PAGE_SCAN_REP_MODE_CHANGE 0x0000000080000000
- #define NG_HCI_FILTER_TYPE_NONE 0x00
- #define NG_HCI_FILTER_TYPE_INQUIRY_RESULT 0x01
- #define NG_HCI_FILTER_TYPE_CON_SETUP 0x02
- #define NG_HCI_FILTER_COND_INQUIRY_NEW_UNIT 0x00
- #define NG_HCI_FILTER_COND_INQUIRY_UNIT_CLASS 0x01
- #define NG_HCI_FILTER_COND_INQUIRY_BDADDR 0x02
- #define NG_HCI_FILTER_COND_CON_ANY_UNIT 0x00
- #define NG_HCI_FILTER_COND_CON_UNIT_CLASS 0x01
- #define NG_HCI_FILTER_COND_CON_BDADDR 0x02
- #define NG_HCI_XMIT_LEVEL_CURRENT 0x00
- #define NG_HCI_XMIT_LEVEL_MAXIMUM 0x01
- #define NG_HCI_H2HC_FLOW_CONTROL_NONE 0x00
- #define NG_HCI_H2HC_FLOW_CONTROL_ACL 0x01
- #define NG_HCI_H2HC_FLOW_CONTROL_SCO 0x02
- #define NG_HCI_H2HC_FLOW_CONTROL_BOTH 0x03
- #define NG_HCI_COUNTRY_CODE_NAM_EUR_JP 0x00

- #define NG_HCI_COUNTRY_CODE_FRANCE 0x01
- #define NG_HCI_LOOPBACK_NONE 0x00
- #define NG_HCI_LOOPBACK_LOCAL 0x01
- #define NG_HCI_LOOPBACK_REMOTE 0x02
- #define NG_HCI_OPCODE(gf, cf) (((gf) & 0x3f) << 10) | ((cf) & 0x3ff)
- #define NG_HCI_OCF(op) ((op) & 0x3ff)
- #define NG_HCI_OGF(op) (((op) >> 10) & 0x3f)
- #define NG_HCI_CON_HANDLE(h) ((h) & 0x0fff)
- #define NG_HCI_PB_FLAG(h) (((h) & 0x3000) >> 12)
- #define NG_HCI_BC_FLAG(h) (((h) & 0xc000) >> 14)
- #define NG_HCI_MK_CON_HANDLE(h, pb, bc) (((h) & 0x0fff) | (((pb) & 3) << 12) | (((bc) & 3) << 14))
- #define NG_HCI_PACKET_FRAGMENT 0x1
- #define NG_HCI_PACKET_START 0x2
- #define NG_HCI_POINT2POINT 0x0
- #define NG_HCI_BROADCAST_ACTIVE 0x1
- #define NG_HCI_BROADCAST_PICONET 0x2
- #define NG_HCI_CMD_PKT 0x01
- #define NG_HCI_CMD_PKT_SIZE 0xff
- #define NG_HCI_ACL_DATA_PKT 0x02
- #define NG_HCI_ACL_PKT_SIZE 0xffff
- #define NG_HCI_SCO_DATA_PKT 0x03
- #define NG_HCI_SCO_PKT_SIZE 0xff
- #define NG_HCI_EVENT_PKT 0x04
- #define NG_HCI_EVENT_PKT_SIZE 0xff
- #define NG_HCI_BDADDR_ANY ((bdaddr_p) "\000\000\000\000\000\000")
- #define NGM_HCI_LP_CON_REQ 1
- #define NGM_HCI_LP_DISCON_REQ 2
- #define NGM_HCI_LP_CON_CFM 3
- #define NGM_HCI_LP_CON_IND 4
- #define NGM_HCI_LP_CON_RSP 5
- #define NGM_HCI_LP_DISCON_IND 6
- #define NGM_HCI_LP_QOS_REQ 7
- #define NGM_HCI_LP_QOS_CFM 8
- #define NGM_HCI_LP_QOS_IND 9
- #define NG_HCI_ALERT_LEVEL 1
- #define NG_HCI_ERR_LEVEL 2
- #define NG_HCI_WARN_LEVEL 3
- #define NG_HCI_INFO_LEVEL 4
- #define NG_HCI_UNIT_CONNECTED (1 << 0)
- #define NG_HCI_UNIT_INITED (1 << 1)
- #define NG_HCI_UNIT_READY (NG_HCI_UNIT_CONNECTED|NG_HCI_UNIT_INITED)
- #define NG_HCI_UNIT_COMMAND_PENDING (1 << 2)
- #define NG_HCI_CON_CLOSED 0
- #define NG_HCI_CON_W4_LP_CON_RSP 1
- #define NG_HCI_CON_W4_CONN_COMPLETE 2
- #define NG_HCI_CON_OPEN 3
- #define NGM_HCI_NODE_GET_STATE 100
- #define NGM_HCI_NODE_INIT 101
- #define NGM_HCI_NODE_GET_DEBUG 102

- #define `NGM_HCI_NODE_SET_DEBUG` 103
- #define `NGM_HCI_NODE_GET_BUFFER` 104
- #define `NGM_HCI_NODE_GET_BDADDR` 105
- #define `NGM_HCI_NODE_GET_FEATURES` 106
- #define `NGM_HCI_NODE_GET_STAT` 107
- #define `NGM_HCI_NODE_RESET_STAT` 108
- #define `NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE` 109
- #define `NGM_HCI_NODE_GET_NEIGHBOR_CACHE` 110
- #define `NG_HCI_MAX_NEIGHBOR_NUM` $((0xffff - \text{sizeof}(\text{ng_hci_node_get_neighbor_cache_ep}))/\text{sizeof}(\text{ng_hci_node_neighbor_cache_entry_ep}))$
- #define `NGM_HCI_NODE_GET_CON_LIST` 111
- #define `NG_HCI_MAX_CON_NUM` $((0xffff - \text{sizeof}(\text{ng_hci_node_con_list_ep}))/\text{sizeof}(\text{ng_hci_node_con_ep}))$
- #define `NGM_HCI_NODE_UP` 112
- #define `NGM_HCI_SYNC_CON_QUEUE` 113
- #define `NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK` 114
- #define `NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK` 115
- #define `NGM_HCI_NODE_GET_PACKET_MASK` 116
- #define `NGM_HCI_NODE_SET_PACKET_MASK` 117
- #define `NGM_HCI_NODE_GET_ROLE_SWITCH` 118
- #define `NGM_HCI_NODE_SET_ROLE_SWITCH` 119
- #define `NGM_HCI_NODE_LIST_NAMES` 200
- #define `NG_HCI_OGF_LINK_CONTROL` 0x01
- #define `NG_HCI_OCF_INQUIRY` 0x0001
- #define `NG_HCI_OCF_INQUIRY_CANCEL` 0x0002
- #define `NG_HCI_OCF_PERIODIC_INQUIRY` 0x0003
- #define `NG_HCI_OCF_EXIT_PERIODIC_INQUIRY` 0x0004
- #define `NG_HCI_OCF_CREATE_CON` 0x0005
- #define `NG_HCI_OCF_DISCON` 0x0006
- #define `NG_HCI_OCF_ADD_SCO_CON` 0x0007
- #define `NG_HCI_OCF_ACCEPT_CON` 0x0009
- #define `NG_HCI_OCF_REJECT_CON` 0x000a
- #define `NG_HCI_OCF_LINK_KEY_REP` 0x000b
- #define `NG_HCI_OCF_LINK_KEY_NEG_REP` 0x000c
- #define `NG_HCI_OCF_PIN_CODE_REP` 0x000d
- #define `NG_HCI_OCF_PIN_CODE_NEG_REP` 0x000e
- #define `NG_HCI_OCF_CHANGE_CON_PKT_TYPE` 0x000f
- #define `NG_HCI_OCF_AUTH_REQ` 0x0011
- #define `NG_HCI_OCF_SET_CON_ENCRYPTION` 0x0013
- #define `NG_HCI_OCF_CHANGE_CON_LINK_KEY` 0x0015
- #define `NG_HCI_OCF_MASTER_LINK_KEY` 0x0017
- #define `NG_HCI_OCF_REMOTE_NAME_REQ` 0x0019
- #define `NG_HCI_OCF_READ_REMOTE_FEATURES` 0x001b
- #define `NG_HCI_OCF_READ_REMOTE_VER_INFO` 0x001d
- #define `NG_HCI_OCF_READ_CLOCK_OFFSET` 0x001f
- #define `NG_HCI_OGF_LINK_POLICY` 0x02
- #define `NG_HCI_OCF_HOLD_MODE` 0x0001
- #define `NG_HCI_OCF_SNIFF_MODE` 0x0003
- #define `NG_HCI_OCF_EXIT_SNIFF_MODE` 0x0004
- #define `NG_HCI_OCF_PARK_MODE` 0x0005

- #define NG_HCI_OCF_EXIT_PARK_MODE 0x0006
- #define NG_HCI_OCF_QOS_SETUP 0x0007
- #define NG_HCI_OCF_ROLE_DISCOVERY 0x0009
- #define NG_HCI_OCF_SWITCH_ROLE 0x000b
- #define NG_HCI_OCF_READ_LINK_POLICY_SETTINGS 0x000c
- #define NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS 0x000d
- #define NG_HCI_OGF_HC_BASEBAND 0x03
- #define NG_HCI_OCF_SET_EVENT_MASK 0x0001
- #define NG_HCI_OCF_RESET 0x0003
- #define NG_HCI_OCF_SET_EVENT_FILTER 0x0005
- #define NG_HCI_OCF_FLUSH 0x0008
- #define NG_HCI_OCF_READ_PIN_TYPE 0x0009
- #define NG_HCI_OCF_WRITE_PIN_TYPE 0x000a
- #define NG_HCI_OCF_CREATE_NEW_UNIT_KEY 0x000b
- #define NG_HCI_OCF_READ_STORED_LINK_KEY 0x000d
- #define NG_HCI_OCF_WRITE_STORED_LINK_KEY 0x0011
- #define NG_HCI_OCF_DELETE_STORED_LINK_KEY 0x0012
- #define NG_HCI_OCF_CHANGE_LOCAL_NAME 0x0013
- #define NG_HCI_OCF_READ_LOCAL_NAME 0x0014
- #define NG_HCI_OCF_READ_CON_ACCEPT_TIMO 0x0015
- #define NG_HCI_OCF_WRITE_CON_ACCEPT_TIMO 0x0016
- #define NG_HCI_OCF_READ_PAGE_TIMO 0x0017
- #define NG_HCI_OCF_WRITE_PAGE_TIMO 0x0018
- #define NG_HCI_OCF_READ_SCAN_ENABLE 0x0019
- #define NG_HCI_OCF_WRITE_SCAN_ENABLE 0x001a
- #define NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY 0x001b
- #define NG_HCI_OCF_WRITE_PAGE_SCAN_ACTIVITY 0x001c
- #define NG_HCI_OCF_READ_INQUIRY_SCAN_ACTIVITY 0x001d
- #define NG_HCI_OCF_WRITE_INQUIRY_SCAN_ACTIVITY 0x001e
- #define NG_HCI_OCF_READ_AUTH_ENABLE 0x001f
- #define NG_HCI_OCF_WRITE_AUTH_ENABLE 0x0020
- #define NG_HCI_OCF_READ_ENCRYPTION_MODE 0x0021
- #define NG_HCI_OCF_WRITE_ENCRYPTION_MODE 0x0022
- #define NG_HCI_OCF_READ_UNIT_CLASS 0x0023
- #define NG_HCI_OCF_WRITE_UNIT_CLASS 0x0024
- #define NG_HCI_OCF_READ_VOICE_SETTINGS 0x0025
- #define NG_HCI_OCF_WRITE_VOICE_SETTINGS 0x0026
- #define NG_HCI_OCF_READ_AUTO_FLUSH_TIMO 0x0027
- #define NG_HCI_OCF_WRITE_AUTO_FLUSH_TIMO 0x0028
- #define NG_HCI_OCF_READ_NUM_BROADCAST_RETRANS 0x0029
- #define NG_HCI_OCF_WRITE_NUM_BROADCAST_RETRANS 0x002a
- #define NG_HCI_OCF_READ_HOLD_MODE_ACTIVITY 0x002b
- #define NG_HCI_OCF_WRITE_HOLD_MODE_ACTIVITY 0x002c
- #define NG_HCI_OCF_READ_XMIT_LEVEL 0x002d
- #define NG_HCI_OCF_READ_SCO_FLOW_CONTROL 0x002e
- #define NG_HCI_OCF_WRITE_SCO_FLOW_CONTROL 0x002f
- #define NG_HCI_OCF_H2HC_FLOW_CONTROL 0x0031
- #define NG_HCI_OCF_HOST_BUFFER_SIZE 0x0033
- #define NG_HCI_OCF_HOST_NUM_COMPL_PKTS 0x0035
- #define NG_HCI_OCF_READ_LINK_SUPERVISION_TIMO 0x0036

- #define NG_HCI_OCF_WRITE_LINK_SUPERVISION_TIMO 0x0037
- #define NG_HCI_OCF_READ_SUPPORTED_IAC_NUM 0x0038
- #define NG_HCI_OCF_READ_IAC_LAP 0x0039
- #define NG_HCI_OCF_WRITE_IAC_LAP 0x003a
- #define NG_HCI_OCF_READ_PAGE_SCAN_PERIOD 0x003b
- #define NG_HCI_OCF_WRITE_PAGE_SCAN_PERIOD 0x003c
- #define NG_HCI_OCF_READ_PAGE_SCAN 0x003d
- #define NG_HCI_OCF_WRITE_PAGE_SCAN 0x003e
- #define NG_HCI_OGF_INFO 0x04
- #define NG_HCI_OCF_READ_LOCAL_VER 0x0001
- #define NG_HCI_OCF_READ_LOCAL_FEATURES 0x0003
- #define NG_HCI_OCF_READ_BUFFER_SIZE 0x0005
- #define NG_HCI_OCF_READ_COUNTRY_CODE 0x0007
- #define NG_HCI_OCF_READ_BDADDR 0x0009
- #define NG_HCI_OGF_STATUS 0x05
- #define NG_HCI_OCF_READ_FAILED_CONTACT_CNTR 0x0001
- #define NG_HCI_OCF_RESET_FAILED_CONTACT_CNTR 0x0002
- #define NG_HCI_OCF_GET_LINK_QUALITY 0x0003
- #define NG_HCI_OCF_READ_RSSI 0x0005
- #define NG_HCI_OGF_TESTING 0x06
- #define NG_HCI_OCF_READ_LOOPBACK_MODE 0x0001
- #define NG_HCI_OCF_WRITE_LOOPBACK_MODE 0x0002
- #define NG_HCI_OCF_ENABLE_UNIT_UNDER_TEST 0x0003
- #define NG_HCI_OGF_BT_LOGO 0x3e
- #define NG_HCI_OGF_VENDOR 0x3f
- #define NG_HCI_EVENT_INQUIRY_COMPL 0x01
- #define NG_HCI_EVENT_INQUIRY_RESULT 0x02
- #define NG_HCI_EVENT_CON_COMPL 0x03
- #define NG_HCI_EVENT_CON_REQ 0x04
- #define NG_HCI_EVENT_DISCON_COMPL 0x05
- #define NG_HCI_EVENT_AUTH_COMPL 0x06
- #define NG_HCI_EVENT_REMOTE_NAME_REQ_COMPL 0x07
- #define NG_HCI_EVENT_ENCRYPTION_CHANGE 0x08
- #define NG_HCI_EVENT_CHANGE_CON_LINK_KEY_COMPL 0x09
- #define NG_HCI_EVENT_MASTER_LINK_KEY_COMPL 0x0a
- #define NG_HCI_EVENT_READ_REMOTE_FEATURES_COMPL 0x0b
- #define NG_HCI_EVENT_READ_REMOTE_VER_INFO_COMPL 0x0c
- #define NG_HCI_EVENT_QOS_SETUP_COMPL 0x0d
- #define NG_HCI_EVENT_COMMAND_COMPL 0x0e
- #define NG_HCI_EVENT_COMMAND_STATUS 0x0f
- #define NG_HCI_EVENT_HARDWARE_ERROR 0x10
- #define NG_HCI_EVENT_FLUSH_OCCUR 0x11
- #define NG_HCI_EVENT_ROLE_CHANGE 0x12
- #define NG_HCI_EVENT_NUM_COMPL_PKTS 0x13
- #define NG_HCI_EVENT_MODE_CHANGE 0x14
- #define NG_HCI_EVENT_RETURN_LINK_KEYS 0x15
- #define NG_HCI_EVENT_PIN_CODE_REQ 0x16
- #define NG_HCI_EVENT_LINK_KEY_REQ 0x17
- #define NG_HCI_EVENT_LINK_KEY_NOTIFICATION 0x18
- #define NG_HCI_EVENT_LOOPBACK_COMMAND 0x19

- #define `NG_HCI_EVENT_DATA_BUFFER_OVERFLOW` 0x1a
- #define `NG_HCI_EVENT_MAX_SLOT_CHANGE` 0x1b
- #define `NG_HCI_EVENT_READ_CLOCK_OFFSET_COMPL` 0x1c
- #define `NG_HCI_EVENT_CON_PKT_TYPE_CHANGED` 0x1d
- #define `NG_HCI_EVENT_QOS_VIOLATION` 0x1e
- #define `NG_HCI_EVENT_PAGE_SCAN_MODE_CHANGE` 0x1f
- #define `NG_HCI_EVENT_PAGE_SCAN_REP_MODE_CHANGE` 0x20
- #define `NG_HCI_EVENT_BT_LOGO` 0xfe
- #define `NG_HCI_EVENT_VENDOR` 0xff

Typedefs

- typedef `bdaddr_t * bdaddr_p`
- typedef `u_int16_t ng_hci_node_state_ep`
- typedef `u_int16_t ng_hci_node_debug_ep`
- typedef `u_int16_t ng_hci_node_link_policy_mask_ep`
- typedef `u_int16_t ng_hci_node_packet_mask_ep`
- typedef `u_int16_t ng_hci_node_role_switch_ep`
- typedef `ng_hci_status_rp ng_hci_inquiry_cancel_rp`
- typedef `ng_hci_status_rp ng_hci_periodic_inquiry_rp`
- typedef `ng_hci_status_rp ng_hci_exit_periodic_inquiry_rp`
- typedef `ng_hci_status_rp ng_hci_set_event_mask_rp`
- typedef `ng_hci_status_rp ng_hci_reset_rp`
- typedef `ng_hci_status_rp ng_hci_set_event_filter_rp`
- typedef `ng_hci_status_rp ng_hci_write_pin_type_rp`
- typedef `ng_hci_status_rp ng_hci_create_new_unit_key_rp`
- typedef `ng_hci_status_rp ng_hci_change_local_name_rp`
- typedef `ng_hci_status_rp ng_hci_write_con_accept_timo_rp`
- typedef `ng_hci_status_rp ng_hci_write_page_timo_rp`
- typedef `ng_hci_status_rp ng_hci_write_scan_enable_rp`
- typedef `ng_hci_status_rp ng_hci_write_page_scan_activity_rp`
- typedef `ng_hci_status_rp ng_hci_write_inquiry_scan_activity_rp`
- typedef `ng_hci_status_rp ng_hci_write_auth_enable_rp`
- typedef `ng_hci_status_rp ng_hci_write_encryption_mode_rp`
- typedef `ng_hci_status_rp ng_hci_write_unit_class_rp`
- typedef `ng_hci_status_rp ng_hci_write_voice_settings_rp`
- typedef `ng_hci_status_rp ng_hci_write_num_broadcast_retrans_rp`
- typedef `ng_hci_status_rp ng_hci_write_hold_mode_activity_rp`
- typedef `ng_hci_status_rp ng_hci_write_sco_flow_control_rp`
- typedef `ng_hci_status_rp ng_hci_h2hc_flow_control_rp`
- typedef `ng_hci_status_rp ng_hci_host_buffer_size_rp`
- typedef `ng_hci_status_rp ng_hci_write_iac_lap_rp`
- typedef `ng_hci_status_rp ng_hci_write_page_scan_period_rp`
- typedef `ng_hci_status_rp ng_hci_write_page_scan_rp`
- typedef `ng_hci_status_rp ng_hci_write_loopback_mode_rp`
- typedef `ng_hci_status_rp ng_hci_enable_unit_under_test_rp`

7.49.1 Define Documentation

7.49.1.1 #define NG_HCI_ACL_DATA_PKT 0x02

Definition at line 347 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_filter(), ng_h4_input(), ng_hci_acl_rcvdata(), ng_hci_drv_rcvdata(), ng_l2cap_lp_send(), ng_ubt_rcvdata(), and ubt_bulk_in_start().

7.49.1.2 #define NG_HCI_ACL_PKT_SIZE 0xffff

Definition at line 348 of file ng_hci.h.

7.49.1.3 #define NG_HCI_ALERT_LEVEL 1

Definition at line 477 of file ng_hci.h.

7.49.1.4 #define NG_HCI_BC_FLAG(h) (((h) & 0xc000) >> 14)

Definition at line 321 of file ng_hci.h.

7.49.1.5 #define NG_HCI_BDADDR_ANY ((bdaddr_p) "\000\000\000\000\000\000")

Definition at line 380 of file ng_hci.h.

Referenced by ng_btsocket_l2cap_connect(), ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_pcb_by_addr(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_raw_bind(), ng_btsocket_l2cap_raw_connect(), ng_btsocket_l2cap_raw_input(), ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_listen(), ng_btsocket_rfcomm_pcb_listener(), ng_btsocket_rfcomm_session_by_addr(), ng_hci_default_rcvmsg(), and ng_l2cap_send_hook_info().

7.49.1.6 #define NG_HCI_BDADDR_SIZE 6

Definition at line 73 of file ng_hci.h.

7.49.1.7 #define NG_HCI_BROADCAST_ACTIVE 0x1

Definition at line 333 of file ng_hci.h.

7.49.1.8 #define NG_HCI_BROADCAST_PICONET 0x2

Definition at line 334 of file ng_hci.h.

7.49.1.9 #define NG_HCI_CLASS_SIZE 3

Definition at line 78 of file ng_hci.h.

Referenced by inquiry_result().

7.49.1.10 #define NG_HCI_CMD_PKT 0x01

Definition at line 338 of file ng_hci.h.

Referenced by con_compl(), ng_btsocket_hci_raw_filter(), ng_btsocket_hci_raw_send(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_raw_rcvdata(), and ng_ubt_rcvdata().

7.49.1.11 #define NG_HCI_CMD_PKT_SIZE 0xff

Definition at line 339 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_send().

7.49.1.12 #define NG_HCI_CON_CLOSED 0

Definition at line 489 of file ng_hci.h.

Referenced by ng_hci_new_con().

7.49.1.13 #define NG_HCI_CON_HANDLE(h) ((h) & 0x0fff)

Definition at line 319 of file ng_hci.h.

Referenced by con_compl(), discon_compl(), encryption_change(), mode_change(), ng_hci_acl_rcvdata(), ng_hci_sco_rcvdata(), ng_l2cap_lp_receive(), num_compl_pkts(), process_link_policy_params(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), and read_remote_features_compl().

7.49.1.14 #define NG_HCI_CON_OPEN 3

Definition at line 492 of file ng_hci.h.

Referenced by con_compl(), ng_hci_acl_rcvdata(), ng_hci_lp_acl_con_req(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_sco_rcvdata(), qos_setup_compl(), and qos_violation().

7.49.1.15 #define NG_HCI_CON_W4_CONN_COMPLETE 2

Definition at line 491 of file ng_hci.h.

Referenced by con_compl(), con_req(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_sco_con_req(), and ng_hci_process_con_timeout().

7.49.1.16 #define NG_HCI_CON_W4_LP_CON_RSP 1

Definition at line 490 of file ng_hci.h.

Referenced by con_req(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_sco_con_req(), and ng_hci_process_con_timeout().

7.49.1.17 #define NG_HCI_COUNTRY_CODE_FRANCE 0x01

Definition at line 290 of file ng_hci.h.

7.49.1.18 #define NG_HCI_COUNTRY_CODE_NAM_EUR_JP 0x00

Definition at line 289 of file ng_hci.h.

7.49.1.19 #define NG_HCI_ENCRYPTION_MODE_ALL 0x02

Definition at line 204 of file ng_hci.h.

7.49.1.20 #define NG_HCI_ENCRYPTION_MODE_NONE 0x00

Definition at line 202 of file ng_hci.h.

Referenced by encryption_change().

7.49.1.21 #define NG_HCI_ENCRYPTION_MODE_P2P 0x01

Definition at line 203 of file ng_hci.h.

Referenced by encryption_change().

7.49.1.22 #define NG_HCI_ERR_LEVEL 2

Definition at line 478 of file ng_hci.h.

7.49.1.23 #define NG_HCI_EVENT_AUTH_COMPL 0x06

Definition at line 1478 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.24 #define NG_HCI_EVENT_BT_LOGO 0xfe

Definition at line 1658 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.25 #define NG_HCI_EVENT_CHANGE_CON_LINK_KEY_COMPL 0x09

Definition at line 1498 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.26 #define NG_HCI_EVENT_COMMAND_COMPL 0x0e

Definition at line 1539 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_attach(), and ng_hci_process_event().

7.49.1.27 #define NG_HCI_EVENT_COMMAND_STATUS 0x0f

Definition at line 1546 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_attach(), and ng_hci_process_event().

7.49.1.28 #define NG_HCI_EVENT_CON_COMPL 0x03

Definition at line 1455 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.29 #define NG_HCI_EVENT_CON_PKT_TYPE_CHANGED 0x1d

Definition at line 1634 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.30 #define NG_HCI_EVENT_CON_REQ 0x04

Definition at line 1464 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.31 #define NG_HCI_EVENT_DATA_BUFFER_OVERFLOW 0x1a

Definition at line 1616 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.32 #define NG_HCI_EVENT_DISCON_COMPL 0x05

Definition at line 1471 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.33 #define NG_HCI_EVENT_ENCRYPTION_CHANGE 0x08

Definition at line 1491 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.34 #define NG_HCI_EVENT_FLUSH_OCCUR 0x11

Definition at line 1558 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.35 #define NG_HCI_EVENT_HARDWARE_ERROR 0x10

Definition at line 1553 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.36 #define NG_HCI_EVENT_INQUIRY_COMPL 0x01

Definition at line 1435 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.37 #define NG_HCI_EVENT_INQUIRY_RESULT 0x02

Definition at line 1440 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.38 #define NG_HCI_EVENT_LINK_KEY_NOTIFICATION 0x18

Definition at line 1604 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and ng_hci_process_event().

7.49.1.39 #define NG_HCI_EVENT_LINK_KEY_REQ 0x17

Definition at line 1599 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.40 #define NG_HCI_EVENT_LOOPBACK_COMMAND 0x19

Definition at line 1611 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.41 #define NG_HCI_EVENT_MASK_SIZE 8

Definition at line 77 of file ng_hci.h.

7.49.1.42 #define NG_HCI_EVENT_MASTER_LINK_KEY_COMPL 0x0a

Definition at line 1504 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.43 #define NG_HCI_EVENT_MAX_SLOT_CHANGE 0x1b

Definition at line 1621 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.44 #define NG_HCI_EVENT_MODE_CHANGE 0x14

Definition at line 1578 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.45 #define NG_HCI_EVENT_NUM_COMPL_PKTS 0x13

Definition at line 1570 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.46 #define NG_HCI_EVENT_PAGE_SCAN_MODE_CHANGE 0x1f

Definition at line 1646 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.47 #define NG_HCI_EVENT_PAGE_SCAN_REP_MODE_CHANGE 0x20

Definition at line 1652 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.48 #define NG_HCI_EVENT_PIN_CODE_REQ 0x16

Definition at line 1594 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.49 #define NG_HCI_EVENT_PKT 0x04

Definition at line 365 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_filter(), ng_h4_input(), ng_hci_drv_rcvdata(), and ubt_intr_start().

7.49.1.50 #define NG_HCI_EVENT_PKT_SIZE 0xff

Definition at line 366 of file ng_hci.h.

7.49.1.51 #define NG_HCI_EVENT_QOS_SETUP_COMPL 0x0d

Definition at line 1527 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.52 #define NG_HCI_EVENT_QOS_VIOLATION 0x1e

Definition at line 1641 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.53 #define NG_HCI_EVENT_READ_CLOCK_OFFSET_COMPL 0x1c

Definition at line 1627 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.54 #define NG_HCI_EVENT_READ_REMOTE_FEATURES_COMPL 0x0b

Definition at line 1511 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.55 #define NG_HCI_EVENT_READ_REMOTE_VER_INFO_COMPL 0x0c

Definition at line 1518 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.56 #define NG_HCI_EVENT_REMOTE_NAME_REQ_COMPL 0x7

Definition at line 1484 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.57 #define NG_HCI_EVENT_RETURN_LINK_KEYS 0x15

Definition at line 1586 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and ng_hci_process_event().

7.49.1.58 #define NG_HCI_EVENT_ROLE_CHANGE 0x12

Definition at line 1563 of file ng_hci.h.

Referenced by ng_hci_process_event().

7.49.1.59 #define NG_HCI_EVENT_VENDOR 0xff

Definition at line 1660 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and ng_hci_process_event().

7.49.1.60 #define NG_HCI_EVMSK_ALL 0x00000000ffffff

Definition at line 222 of file ng_hci.h.

7.49.1.61 #define NG_HCI_EVMSK_AUTH_COMPL 0x0000000000000020

Definition at line 229 of file ng_hci.h.

**7.49.1.62 #define NG_HCI_EVMSK_CHANGE_CON_LINK_KEY_-
COMPL 0x0000000000000100**

Definition at line 232 of file ng_hci.h.

7.49.1.63 #define NG_HCI_EVMSK_COMMAND_COMPL 0x0000000000002000

Definition at line 237 of file ng_hci.h.

7.49.1.64 #define NG_HCI_EVMSK_COMMAND_STATUS 0x0000000000004000

Definition at line 238 of file ng_hci.h.

7.49.1.65 #define NG_HCI_EVMSK_CON_COMPL 0x0000000000000004

Definition at line 226 of file ng_hci.h.

7.49.1.66 #define NG_HCI_EVMSK_CON_PKT_TYPE_CHANGED 0x0000000010000000

Definition at line 252 of file ng_hci.h.

7.49.1.67 #define NG_HCI_EVMSK_CON_REQ 0x0000000000000008

Definition at line 227 of file ng_hci.h.

7.49.1.68 #define NG_HCI_EVMSK_DATA_BUFFER_OVERFLOW 0x0000000002000000

Definition at line 249 of file ng_hci.h.

7.49.1.69 #define NG_HCI_EVMSK_DISCON_COMPL 0x0000000000000010

Definition at line 228 of file ng_hci.h.

7.49.1.70 #define NG_HCI_EVMSK_ENCRYPTION_CHANGE 0x0000000000000080

Definition at line 231 of file ng_hci.h.

7.49.1.71 #define NG_HCI_EVMSK_FLUSH_OCCUR 0x0000000000001000

Definition at line 240 of file ng_hci.h.

7.49.1.72 #define NG_HCI_EVMSK_HARDWARE_ERROR 0x0000000000008000

Definition at line 239 of file ng_hci.h.

7.49.1.73 #define NG_HCI_EVMSK_INQUIRY_COMPL 0x0000000000000001

Definition at line 224 of file ng_hci.h.

7.49.1.74 #define NG_HCI_EVMSK_INQUIRY_RESULT 0x0000000000000002

Definition at line 225 of file ng_hci.h.

7.49.1.75 #define NG_HCI_EVMSK_LINK_KEY_NOTIFICATION 0x0000000000800000

Definition at line 247 of file ng_hci.h.

7.49.1.76 #define NG_HCI_EVMSK_LINK_KEY_REQ 0x0000000000400000

Definition at line 246 of file ng_hci.h.

7.49.1.77 #define NG_HCI_EVMSK_LOOPBACK_COMMAND 0x0000000001000000

Definition at line 248 of file ng_hci.h.

7.49.1.78 #define NG_HCI_EVMSK_MASTER_LINK_KEY_COMPL 0x0000000000000200

Definition at line 233 of file ng_hci.h.

7.49.1.79 #define NG_HCI_EVMSK_MAX_SLOT_CHANGE 0x0000000004000000

Definition at line 250 of file ng_hci.h.

7.49.1.80 #define NG_HCI_EVMSK_MODE_CHANGE 0x0000000000800000

Definition at line 243 of file ng_hci.h.

7.49.1.81 #define NG_HCI_EVMSK_NONE 0x0000000000000000

Definition at line 223 of file ng_hci.h.

7.49.1.82 #define NG_HCI_EVMSK_NUM_COMPL_PKTS 0x00000000040000

Definition at line 242 of file ng_hci.h.

7.49.1.83 #define NG_HCI_EVMSK_PAGE_SCAN_MODE_CHANGE 0x0000000040000000

Definition at line 254 of file ng_hci.h.

**7.49.1.84 #define NG_HCI_EVMSK_PAGE_SCAN_REP_MODE_-
CHANGE 0x0000000080000000**

Definition at line 255 of file ng_hci.h.

7.49.1.85 #define NG_HCI_EVMSK_PIN_CODE_REQ 0x000000000200000

Definition at line 245 of file ng_hci.h.

7.49.1.86 #define NG_HCI_EVMSK_QOS_SETUP_COMPL 0x000000000001000

Definition at line 236 of file ng_hci.h.

7.49.1.87 #define NG_HCI_EVMSK_QOS_VIOLATION 0x000000002000000

Definition at line 253 of file ng_hci.h.

**7.49.1.88 #define NG_HCI_EVMSK_READ_CLOCK_OFFSET_-
COMPLETE 0x000000000800000**

Definition at line 251 of file ng_hci.h.

**7.49.1.89 #define NG_HCI_EVMSK_READ_REMOTE_FEATURES_-
COMPL 0x000000000000400**

Definition at line 234 of file ng_hci.h.

**7.49.1.90 #define NG_HCI_EVMSK_READ_REMOTE_VER_INFO_-
COMPL 0x000000000000800**

Definition at line 235 of file ng_hci.h.

7.49.1.91 #define NG_HCI_EVMSK_REMOTE_NAME_REQ_COMPL 0x000000000000040

Definition at line 230 of file ng_hci.h.

7.49.1.92 #define NG_HCI_EVMSK_RETURN_LINK_KEYS 0x000000000100000

Definition at line 244 of file ng_hci.h.

7.49.1.93 #define NG_HCI_EVMSK_ROLE_CHANGE 0x000000000020000

Definition at line 241 of file ng_hci.h.

7.49.1.94 #define NG_HCI_FEATURES_SIZE 8

Definition at line 79 of file ng_hci.h.

7.49.1.95 #define NG_HCI_FILTER_COND_CON_ANY_UNIT 0x00

Definition at line 271 of file ng_hci.h.

7.49.1.96 #define NG_HCI_FILTER_COND_CON_BDADDR 0x02

Definition at line 273 of file ng_hci.h.

7.49.1.97 #define NG_HCI_FILTER_COND_CON_UNIT_CLASS 0x01

Definition at line 272 of file ng_hci.h.

7.49.1.98 #define NG_HCI_FILTER_COND_INQUIRY_BDADDR 0x02

Definition at line 267 of file ng_hci.h.

7.49.1.99 #define NG_HCI_FILTER_COND_INQUIRY_NEW_UNIT 0x00

Definition at line 265 of file ng_hci.h.

7.49.1.100 #define NG_HCI_FILTER_COND_INQUIRY_UNIT_CLASS 0x01

Definition at line 266 of file ng_hci.h.

7.49.1.101 #define NG_HCI_FILTER_TYPE_CON_SETUP 0x02

Definition at line 261 of file ng_hci.h.

7.49.1.102 #define NG_HCI_FILTER_TYPE_INQUIRY_RESULT 0x01

Definition at line 260 of file ng_hci.h.

7.49.1.103 #define NG_HCI_FILTER_TYPE_NONE 0x00

Definition at line 259 of file ng_hci.h.

7.49.1.104 #define NG_HCI_H2HC_FLOW_CONTROL_ACL 0x01

Definition at line 283 of file ng_hci.h.

7.49.1.105 #define NG_HCI_H2HC_FLOW_CONTROL_BOTH 0x03

Definition at line 285 of file ng_hci.h.

7.49.1.106 #define NG_HCI_H2HC_FLOW_CONTROL_NONE 0x00

Definition at line 282 of file ng_hci.h.

7.49.1.107 #define NG_HCI_H2HC_FLOW_CONTROL_SCO 0x02

Definition at line 284 of file ng_hci.h.

7.49.1.108 #define NG_HCI_HOLD_MODE_NO_CHANGE 0x00

Definition at line 175 of file ng_hci.h.

7.49.1.109 #define NG_HCI_HOLD_MODE_SUSPEND_INQUIRY_SCAN 0x02

Definition at line 177 of file ng_hci.h.

7.49.1.110 #define NG_HCI_HOLD_MODE_SUSPEND_PAGE_SCAN 0x01

Definition at line 176 of file ng_hci.h.

7.49.1.111 #define NG_HCI_HOLD_MODE_SUSPEND_PERIOD_INQUIRY 0x04

Definition at line 178 of file ng_hci.h.

7.49.1.112 #define NG_HCI_HOOK_ACL "acl"

Definition at line 62 of file ng_hci.h.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_newhook().

7.49.1.113 #define NG_HCI_HOOK_DRV "drv"

Definition at line 61 of file ng_hci.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_newhook(), ng_hci_send_command(), and send_data_packets().

7.49.1.114 #define NG_HCI_HOOK_RAW "raw"

Definition at line 64 of file ng_hci.h.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_newhook().

7.49.1.115 #define NG_HCI_HOOK_SCO "sco"

Definition at line 63 of file ng_hci.h.

Referenced by ng_hci_default_rcvmsg(), and ng_hci_newhook().

7.49.1.116 #define NG_HCI_INFO_LEVEL 4

Definition at line 480 of file ng_hci.h.

7.49.1.117 #define NG_HCI_INQUIRY_DISABLE_PAGE_ENABLE 0x02

Definition at line 170 of file ng_hci.h.

7.49.1.118 #define NG_HCI_INQUIRY_ENABLE_PAGE_DISABLE 0x01

Definition at line 169 of file ng_hci.h.

7.49.1.119 #define NG_HCI_INQUIRY_ENABLE_PAGE_ENABLE 0x03

Definition at line 171 of file ng_hci.h.

7.49.1.120 #define NG_HCI_KEY_SIZE 16

Definition at line 75 of file ng_hci.h.

7.49.1.121 #define NG_HCI_LAP_SIZE 3

Definition at line 74 of file ng_hci.h.

7.49.1.122 #define NG_HCI_LINK_ACL 0x01

Definition at line 117 of file ng_hci.h.

Referenced by con_compl(), data_buffer_overflow(), encryption_change(), mode_change(), ng_hci_acl_rcvdata(), ng_hci_free_con(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_ind(), ng_hci_lp_con_req(), ng_hci_lp_discon_ind(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_new_con(), ng_hci_send_data(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), num_compl_pkts(), process_link_policy_params(), qos_setup_compl(), qos_violation(), role_change(), and sync_con_queue().

7.49.1.123 #define NG_HCI_LINK_KEY_TYPE_COMBINATION_KEY 0x00

Definition at line 196 of file ng_hci.h.

7.49.1.124 #define NG_HCI_LINK_KEY_TYPE_LOCAL_UNIT_KEY 0x01

Definition at line 197 of file ng_hci.h.

7.49.1.125 #define NG_HCI_LINK_KEY_TYPE_REMOTE_UNIT_KEY 0x02

Definition at line 198 of file ng_hci.h.

7.49.1.126 #define NG_HCI_LINK_POLICY_DISABLE_ALL_LM_MODES 0x0000

Definition at line 214 of file ng_hci.h.

7.49.1.127 #define NG_HCI_LINK_POLICY_ENABLE_HOLD_MODE 0x0002

Definition at line 216 of file ng_hci.h.

7.49.1.128 #define NG_HCI_LINK_POLICY_ENABLE_PARK_MODE 0x0008

Definition at line 218 of file ng_hci.h.

7.49.1.129 #define NG_HCI_LINK_POLICY_ENABLE_ROLE_SWITCH 0x0001

Definition at line 215 of file ng_hci.h.

7.49.1.130 #define NG_HCI_LINK_POLICY_ENABLE_SNIFF_MODE 0x0004

Definition at line 217 of file ng_hci.h.

7.49.1.131 #define NG_HCI_LINK_SCO 0x00

Definition at line 116 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req(), ng_hci_sco_rcvdata(), and ng_hci_send_data().

7.49.1.132 #define NG_HCI_LMP_3SLOT 0x01

Definition at line 89 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.133 #define NG_HCI_LMP_5SLOT 0x02

Definition at line 90 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.134 #define NG_HCI_LMP_ALAW_LOG 0x80

Definition at line 105 of file ng_hci.h.

7.49.1.135 #define NG_HCI_LMP_CHANNEL_QUALITY 0x04

Definition at line 100 of file ng_hci.h.

7.49.1.136 #define NG_HCI_LMP_CVSD 0x01

Definition at line 107 of file ng_hci.h.

7.49.1.137 #define NG_HCI_LMP_ENCRYPTION 0x04

Definition at line 91 of file ng_hci.h.

7.49.1.138 #define NG_HCI_LMP_FLOW_CONTROL_LAG0 0x10

Definition at line 111 of file ng_hci.h.

7.49.1.139 #define NG_HCI_LMP_FLOW_CONTROL_LAG1 0x20

Definition at line 112 of file ng_hci.h.

7.49.1.140 #define NG_HCI_LMP_FLOW_CONTROL_LAG2 0x40

Definition at line 113 of file ng_hci.h.

7.49.1.141 #define NG_HCI_LMP_HOLD_MODE 0x40

Definition at line 95 of file ng_hci.h.

Referenced by con_compl().

7.49.1.142 #define NG_HCI_LMP_HV2_PKT 0x10

Definition at line 102 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req().

7.49.1.143 #define NG_HCI_LMP_HV3_PKT 0x20

Definition at line 103 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req().

7.49.1.144 #define NG_HCI_LMP_PAGING_SCHEME 0x02

Definition at line 108 of file ng_hci.h.

7.49.1.145 #define NG_HCI_LMP_PARK_MODE 0x01

Definition at line 98 of file ng_hci.h.

Referenced by con_compl().

7.49.1.146 #define NG_HCI_LMP_POWER_CONTROL 0x04

Definition at line 109 of file ng_hci.h.

7.49.1.147 #define NG_HCI_LMP_RSSI 0x02

Definition at line 99 of file ng_hci.h.

7.49.1.148 #define NG_HCI_LMP_SCO_LINK 0x08

Definition at line 101 of file ng_hci.h.

7.49.1.149 #define NG_HCI_LMP_SLOT_OFFSET 0x08

Definition at line 92 of file ng_hci.h.

7.49.1.150 #define NG_HCI_LMP_SNIFF_MODE 0x80

Definition at line 96 of file ng_hci.h.

Referenced by con_compl().

7.49.1.151 #define NG_HCI_LMP_SWITCH 0x20

Definition at line 94 of file ng_hci.h.

Referenced by con_compl(), ng_hci_lp_acl_con_req(), and ng_hci_lp_con_rsp().

7.49.1.152 #define NG_HCI_LMP_TIMING_ACCURACY 0x10

Definition at line 93 of file ng_hci.h.

7.49.1.153 #define NG_HCI_LMP_TRANSPARENT_SCO 0x08

Definition at line 110 of file ng_hci.h.

7.49.1.154 #define NG_HCI_LMP_ULAW_LOG 0x40

Definition at line 104 of file ng_hci.h.

7.49.1.155 #define NG_HCI_LOOPBACK_LOCAL 0x01

Definition at line 295 of file ng_hci.h.

7.49.1.156 #define NG_HCI_LOOPBACK_NONE 0x00

Definition at line 294 of file ng_hci.h.

7.49.1.157 #define NG_HCI_LOOPBACK_REMOTE 0x02

Definition at line 296 of file ng_hci.h.

7.49.1.158 #define NG_HCI_MANDATORY_PAGE_SCAN_MODE 0x00

Definition at line 149 of file ng_hci.h.

7.49.1.159 #define NG_HCI_MAX_CON_NUM ((0xffff - sizeof(ng_hci_node_con_list_ep))/sizeof(ng_hci_node_con_ep))

Definition at line 578 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.160 #define NG_HCI_MAX_NEIGHBOR_NUM ((0xffff - sizeof(ng_hci_node_get_neighbor_cache_ep))/sizeof(ng_hci_node_neighbor_cache_entry_ep))

Definition at line 557 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.161 #define NG_HCI_MK_CON_HANDLE(h, pb, bc) (((h) & 0x0fff) | (((pb) & 3) << 12) | (((bc) & 3) << 14))

Definition at line 322 of file ng_hci.h.

Referenced by ng_l2cap_lp_send().

7.49.1.162 #define NG_HCI_NO_SCAN_ENABLE 0x00

Definition at line 168 of file ng_hci.h.

7.49.1.163 #define NG_HCI_NODE_TYPE "hci"

Definition at line 57 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control().

7.49.1.164 #define NG_HCI_OCF(op) ((op) & 0x3ff)

Definition at line 311 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_filter(), ng_hci_process_command_complete(), ng_hci_process_command_timeout(), process_link_control_status(), and process_link_policy_status().

7.49.1.165 #define NG_HCI_OCF_ACCEPT_CON 0x0009

Definition at line 669 of file ng_hci.h.

Referenced by ng_hci_lp_con_rsp(), process_link_control_params(), and process_link_control_status().

7.49.1.166 #define NG_HCI_OCF_ADD_SCO_CON 0x0007

Definition at line 662 of file ng_hci.h.

Referenced by `ng_hci_lp_sco_con_req()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.167 #define NG_HCI_OCF_AUTH_REQ 0x0011

Definition at line 733 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.168 #define NG_HCI_OCF_CHANGE_CON_LINK_KEY 0x0015

Definition at line 746 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.169 #define NG_HCI_OCF_CHANGE_CON_PKT_TYPE 0x000f

Definition at line 726 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.170 #define NG_HCI_OCF_CHANGE_LOCAL_NAME 0x0013

Definition at line 975 of file `ng_hci.h`.

Referenced by `process_hc_baseband_params()`.

7.49.1.171 #define NG_HCI_OCF_CREATE_CON 0x0005

Definition at line 644 of file `ng_hci.h`.

Referenced by `ng_hci_lp_acl_con_req()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.172 #define NG_HCI_OCF_CREATE_NEW_UNIT_KEY 0x000b

Definition at line 935 of file `ng_hci.h`.

Referenced by `process_hc_baseband_params()`.

7.49.1.173 #define NG_HCI_OCF_DELETE_STORED_LINK_KEY 0x0012

Definition at line 964 of file `ng_hci.h`.

Referenced by `process_hc_baseband_params()`.

7.49.1.174 #define NG_HCI_OCF_DISCON 0x0006

Definition at line 655 of file `ng_hci.h`.

Referenced by `ng_hci_lp_discon_req()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.175 #define NG_HCI_OCF_ENABLE_UNIT_UNDER_TEST 0x0003

Definition at line 1415 of file ng_hci.h.

Referenced by process_testing_params().

7.49.1.176 #define NG_HCI_OCF_EXIT_PARK_MODE 0x0006

Definition at line 825 of file ng_hci.h.

Referenced by process_link_policy_params(), and process_link_policy_status().

7.49.1.177 #define NG_HCI_OCF_EXIT_PERIODIC_INQUIRY 0x0004

Definition at line 640 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_control_params(), and process_link_control_status().

7.49.1.178 #define NG_HCI_OCF_EXIT_SNIFF_MODE 0x0004

Definition at line 811 of file ng_hci.h.

Referenced by process_link_policy_params(), and process_link_policy_status().

7.49.1.179 #define NG_HCI_OCF_FLUSH 0x0008

Definition at line 911 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.180 #define NG_HCI_OCF_GET_LINK_QUALITY 0x0003

Definition at line 1371 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_status_params().

7.49.1.181 #define NG_HCI_OCF_H2HC_FLOW_CONTROL 0x0031

Definition at line 1195 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.182 #define NG_HCI_OCF_HOLD_MODE 0x0001

Definition at line 793 of file ng_hci.h.

Referenced by process_link_policy_params(), and process_link_policy_status().

7.49.1.183 #define NG_HCI_OCF_HOST_BUFFER_SIZE 0x0033

Definition at line 1202 of file ng_hci.h.

Referenced by `process_hc_baseband_params()`.

7.49.1.184 #define NG_HCI_OCF_HOST_NUM_COMPL_PKTS 0x0035

Definition at line 1212 of file `ng_hci.h`.

Referenced by `process_hc_baseband_params()`.

7.49.1.185 #define NG_HCI_OCF_INQUIRY 0x0001

Definition at line 617 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.186 #define NG_HCI_OCF_INQUIRY_CANCEL 0x0002

Definition at line 625 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.187 #define NG_HCI_OCF_LINK_KEY_NEG_REP 0x000c

Definition at line 694 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.188 #define NG_HCI_OCF_LINK_KEY_REP 0x000b

Definition at line 683 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.189 #define NG_HCI_OCF_MASTER_LINK_KEY 0x0017

Definition at line 752 of file `ng_hci.h`.

Referenced by `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.190 #define NG_HCI_OCF_PARK_MODE 0x0005

Definition at line 817 of file `ng_hci.h`.

Referenced by `process_link_policy_params()`, and `process_link_policy_status()`.

7.49.1.191 #define NG_HCI_OCF_PERIODIC_INQUIRY 0x0003

Definition at line 629 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `process_link_control_params()`, and `process_link_control_status()`.

7.49.1.192 #define NG_HCI_OCF_PIN_CODE_NEG_REP 0x000e

Definition at line 716 of file ng_hci.h.

Referenced by process_link_control_params(), and process_link_control_status().

7.49.1.193 #define NG_HCI_OCF_PIN_CODE_REP 0x000d

Definition at line 704 of file ng_hci.h.

Referenced by process_link_control_params(), and process_link_control_status().

7.49.1.194 #define NG_HCI_OCF_QOS_SETUP 0x0007

Definition at line 831 of file ng_hci.h.

Referenced by ng_hci_lp_qos_req(), process_link_policy_params(), and process_link_policy_status().

7.49.1.195 #define NG_HCI_OCF_READ_AUTH_ENABLE 0x001f

Definition at line 1063 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.196 #define NG_HCI_OCF_READ_AUTO_FLUSH_TIMO 0x0027

Definition at line 1119 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.197 #define NG_HCI_OCF_READ_BDADDR 0x0009

Definition at line 1336 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_info_params().

7.49.1.198 #define NG_HCI_OCF_READ_BUFFER_SIZE 0x0005

Definition at line 1321 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_info_params().

7.49.1.199 #define NG_HCI_OCF_READ_CLOCK_OFFSET 0x001f

Definition at line 779 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_control_params(), and process_link_control_status().

7.49.1.200 #define NG_HCI_OCF_READ_CON_ACCEPT_TIMO 0x0015

Definition at line 989 of file ng_hci.h.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.201 #define NG_HCI_OCF_READ_COUNTRY_CODE 0x0007

Definition at line 1330 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_info_params()`.

7.49.1.202 #define NG_HCI_OCF_READ_ENCRYPTION_MODE 0x0021

Definition at line 1077 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.203 #define NG_HCI_OCF_READ_FAILED_CONTACT_CNTR 0x0001

Definition at line 1350 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_status_params()`.

7.49.1.204 #define NG_HCI_OCF_READ_HOLD_MODE_ACTIVITY 0x002b

Definition at line 1155 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.205 #define NG_HCI_OCF_READ_IAC_LAP 0x0039

Definition at line 1250 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.206 #define NG_HCI_OCF_READ_INQUIRY_SCAN_ACTIVITY 0x001d

Definition at line 1047 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.207 #define NG_HCI_OCF_READ_LINK_POLICY_SETTINGS 0x000c

Definition at line 861 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `process_link_policy_params()`, and `process_link_policy_status()`.

7.49.1.208 #define NG_HCI_OCF_READ_LINK_SUPERVISION_TIMO 0x0036

Definition at line 1221 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, and `process_hc_baseband_params()`.

7.49.1.209 #define NG_HCI_OCF_READ_LOCAL_FEATURES 0x0003

Definition at line 1315 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_info_params().

7.49.1.210 #define NG_HCI_OCF_READ_LOCAL_NAME 0x0014

Definition at line 982 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.211 #define NG_HCI_OCF_READ_LOCAL_VER 0x0001

Definition at line 1305 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_info_params().

7.49.1.212 #define NG_HCI_OCF_READ_LOOPBACK_MODE 0x0001

Definition at line 1401 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_testing_params().

7.49.1.213 #define NG_HCI_OCF_READ_NUM_BROADCAST_RETRANS 0x0029

Definition at line 1141 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.214 #define NG_HCI_OCF_READ_PAGE_SCAN 0x003d

Definition at line 1282 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.215 #define NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY 0x001b

Definition at line 1031 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.216 #define NG_HCI_OCF_READ_PAGE_SCAN_PERIOD 0x003b

Definition at line 1268 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.217 #define NG_HCI_OCF_READ_PAGE_TIMO 0x0017

Definition at line 1003 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.218 #define NG_HCI_OCF_READ_PIN_TYPE 0x0009

Definition at line 921 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.219 #define NG_HCI_OCF_READ_REMOTE_FEATURES 0x001b

Definition at line 767 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_control_params(), and process_link_control_status().

7.49.1.220 #define NG_HCI_OCF_READ_REMOTE_VER_INFO 0x001d

Definition at line 773 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_control_params(), and process_link_control_status().

7.49.1.221 #define NG_HCI_OCF_READ_RSSI 0x0005

Definition at line 1382 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_status_params().

7.49.1.222 #define NG_HCI_OCF_READ_SCAN_ENABLE 0x0019

Definition at line 1017 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.223 #define NG_HCI_OCF_READ_SCO_FLOW_CONTROL 0x002e

Definition at line 1181 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.224 #define NG_HCI_OCF_READ_STORED_LINK_KEY 0x000d

Definition at line 939 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.225 #define NG_HCI_OCF_READ_SUPPORTED_IAC_NUM 0x0038

Definition at line 1243 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.226 #define NG_HCI_OCF_READ_UNIT_CLASS 0x0023

Definition at line 1091 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.227 #define NG_HCI_OCF_READ_VOICE_SETTINGS 0x0025

Definition at line 1105 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.228 #define NG_HCI_OCF_READ_XMIT_LEVEL 0x002d

Definition at line 1169 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), and process_hc_baseband_params().

7.49.1.229 #define NG_HCI_OCF_REJECT_CON 0x000a

Definition at line 676 of file ng_hci.h.

Referenced by ng_hci_lp_con_rsp(), process_link_control_params(), and process_link_control_status().

7.49.1.230 #define NG_HCI_OCF_REMOTE_NAME_REQ 0x0019

Definition at line 758 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_control_params(), and process_link_control_status().

7.49.1.231 #define NG_HCI_OCF_RESET 0x0003

Definition at line 898 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.232 #define NG_HCI_OCF_RESET_FAILED_CONTACT_CNTR 0x0002

Definition at line 1361 of file ng_hci.h.

Referenced by process_status_params().

7.49.1.233 #define NG_HCI_OCF_ROLE_DISCOVERY 0x0009

Definition at line 843 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_init(), process_link_policy_params(), and process_link_policy_status().

7.49.1.234 #define NG_HCI_OCF_SET_CON_ENCRYPTION 0x0013

Definition at line 739 of file ng_hci.h.

Referenced by process_link_control_params(), and process_link_control_status().

7.49.1.235 #define NG_HCI_OCF_SET_EVENT_FILTER 0x0005

Definition at line 902 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.236 #define NG_HCI_OCF_SET_EVENT_MASK 0x0001

Definition at line 891 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.237 #define NG_HCI_OCF_SNIFF_MODE 0x0003

Definition at line 801 of file ng_hci.h.

Referenced by process_link_policy_params(), and process_link_policy_status().

7.49.1.238 #define NG_HCI_OCF_SWITCH_ROLE 0x000b

Definition at line 854 of file ng_hci.h.

Referenced by process_link_policy_params(), and process_link_policy_status().

7.49.1.239 #define NG_HCI_OCF_WRITE_AUTH_ENABLE 0x0020

Definition at line 1070 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.240 #define NG_HCI_OCF_WRITE_AUTO_FLUSH_TIMO 0x0028

Definition at line 1130 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.241 #define NG_HCI_OCF_WRITE_CON_ACCEPT_TIMO 0x0016

Definition at line 996 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.242 #define NG_HCI_OCF_WRITE_ENCRYPTION_MODE 0x0022

Definition at line 1084 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.243 #define NG_HCI_OCF_WRITE_HOLD_MODE_ACTIVITY 0x002c

Definition at line 1162 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.244 #define NG_HCI_OCF_WRITE_IAC_LAP 0x003a

Definition at line 1259 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.245 #define NG_HCI_OCF_WRITE_INQUIRY_SCAN_ACTIVITY 0x001e

Definition at line 1055 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.246 #define NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS 0x000d

Definition at line 872 of file ng_hci.h.

Referenced by con_compl(), process_link_policy_params(), and process_link_policy_status().

7.49.1.247 #define NG_HCI_OCF_WRITE_LINK_SUPERVISION_TIMO 0x0037

Definition at line 1232 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.248 #define NG_HCI_OCF_WRITE_LOOPBACK_MODE 0x0002

Definition at line 1408 of file ng_hci.h.

Referenced by process_testing_params().

7.49.1.249 #define NG_HCI_OCF_WRITE_NUM_BROADCAST_RETRANS 0x002a

Definition at line 1148 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.250 #define NG_HCI_OCF_WRITE_PAGE_SCAN 0x003e

Definition at line 1289 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.251 #define NG_HCI_OCF_WRITE_PAGE_SCAN_ACTIVITY 0x001c

Definition at line 1039 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.252 #define NG_HCI_OCF_WRITE_PAGE_SCAN_PERIOD 0x003c

Definition at line 1275 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.253 #define NG_HCI_OCF_WRITE_PAGE_TIMO 0x0018

Definition at line 1010 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.254 #define NG_HCI_OCF_WRITE_PIN_TYPE 0x000a

Definition at line 928 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.255 #define NG_HCI_OCF_WRITE_SCAN_ENABLE 0x001a

Definition at line 1024 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.256 #define NG_HCI_OCF_WRITE_SCO_FLOW_CONTROL 0x002f

Definition at line 1188 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.257 #define NG_HCI_OCF_WRITE_STORED_LINK_KEY 0x0011

Definition at line 951 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.258 #define NG_HCI_OCF_WRITE_UNIT_CLASS 0x0024

Definition at line 1098 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.259 #define NG_HCI_OCF_WRITE_VOICE_SETTINGS 0x0026

Definition at line 1112 of file ng_hci.h.

Referenced by process_hc_baseband_params().

7.49.1.260 #define NG_HCI_OGF(op) (((op) >> 10) & 0x3f)

Definition at line 312 of file ng_hci.h.

Referenced by `ng_btsocket_hci_raw_filter()`, `ng_hci_process_command_complete()`, `ng_hci_process_command_status()`, and `ng_hci_process_command_timeout()`.

7.49.1.261 `#define NG_HCI_OGF_BT_LOGO 0x3e`

Definition at line 1425 of file `ng_hci.h`.

Referenced by `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.262 `#define NG_HCI_OGF_HC_BASEBAND 0x03`

Definition at line 889 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.263 `#define NG_HCI_OGF_INFO 0x04`

Definition at line 1303 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.264 `#define NG_HCI_OGF_LINK_CONTROL 0x01`

Definition at line 615 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_rsp()`, `ng_hci_lp_discon_req()`, `ng_hci_lp_sco_con_req()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.265 `#define NG_HCI_OGF_LINK_POLICY 0x02`

Definition at line 791 of file `ng_hci.h`.

Referenced by `con_compl()`, `ng_btsocket_hci_raw_init()`, `ng_hci_lp_qos_req()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.266 `#define NG_HCI_OGF_STATUS 0x05`

Definition at line 1348 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.267 `#define NG_HCI_OGF_TESTING 0x06`

Definition at line 1399 of file `ng_hci.h`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_hci_process_command_complete()`, and `ng_hci_process_command_status()`.

7.49.1.268 #define NG_HCI_OGF_VENDOR 0x3f

Definition at line 1427 of file ng_hci.h.

Referenced by ng_hci_process_command_complete(), and ng_hci_process_command_status().

7.49.1.269 #define NG_HCI_OPCODE(gf, cf) (((gf) & 0x3f) << 10) | ((cf) & 0x3ff)

Definition at line 310 of file ng_hci.h.

Referenced by con_compl(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), and ng_hci_lp_sco_con_req().

7.49.1.270 #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE1 0x01

Definition at line 150 of file ng_hci.h.

7.49.1.271 #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE2 0x02

Definition at line 151 of file ng_hci.h.

7.49.1.272 #define NG_HCI_OPTIONAL_PAGE_SCAN_MODE3 0x03

Definition at line 152 of file ng_hci.h.

7.49.1.273 #define NG_HCI_PACKET_FRAGMENT 0x1

Definition at line 327 of file ng_hci.h.

Referenced by ng_l2cap_lp_receive(), and ng_l2cap_lp_send().

7.49.1.274 #define NG_HCI_PACKET_START 0x2

Definition at line 328 of file ng_hci.h.

Referenced by ng_l2cap_lp_receive(), and ng_l2cap_lp_send().

7.49.1.275 #define NG_HCI_PAGE_SCAN_PERIOD_MODE0 0x00

Definition at line 162 of file ng_hci.h.

7.49.1.276 #define NG_HCI_PAGE_SCAN_PERIOD_MODE1 0x01

Definition at line 163 of file ng_hci.h.

7.49.1.277 #define NG_HCI_PAGE_SCAN_PERIOD_MODE2 0x02

Definition at line 164 of file ng_hci.h.

7.49.1.278 #define NG_HCI_PB_FLAG(h) (((h) & 0x3000) >> 12)

Definition at line 320 of file ng_hci.h.

Referenced by ng_l2cap_lp_receive().

7.49.1.279 #define NG_HCI_PIN_SIZE 16

Definition at line 76 of file ng_hci.h.

7.49.1.280 #define NG_HCI_PIN_TYPE_FIXED 0x01

Definition at line 193 of file ng_hci.h.

7.49.1.281 #define NG_HCI_PIN_TYPE_VARIABLE 0x00

Definition at line 192 of file ng_hci.h.

7.49.1.282 #define NG_HCI_PKT_DH1 0x0010

Definition at line 123 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.283 #define NG_HCI_PKT_DH3 0x0800

Definition at line 129 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.284 #define NG_HCI_PKT_DH5 0x8000

Definition at line 132 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.285 #define NG_HCI_PKT_DM1 0x0008

Definition at line 122 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.286 #define NG_HCI_PKT_DM3 0x0400

Definition at line 128 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.287 #define NG_HCI_PKT_DM5 0x4000

Definition at line 131 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req().

7.49.1.288 #define NG_HCI_PKT_HV1 0x0020

Definition at line 124 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req().

7.49.1.289 #define NG_HCI_PKT_HV2 0x0040

Definition at line 125 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req().

7.49.1.290 #define NG_HCI_PKT_HV3 0x0080

Definition at line 126 of file ng_hci.h.

Referenced by ng_hci_lp_sco_con_req().

7.49.1.291 #define NG_HCI_POINT2POINT 0x0

Definition at line 332 of file ng_hci.h.

7.49.1.292 #define NG_HCI_ROLE_MASTER 0x00

Definition at line 182 of file ng_hci.h.

Referenced by ng_hci_lp_con_rsp().

7.49.1.293 #define NG_HCI_ROLE_SLAVE 0x01

Definition at line 183 of file ng_hci.h.

Referenced by ng_hci_lp_con_rsp().

7.49.1.294 #define NG_HCI_SCAN_REP_MODE0 0x00

Definition at line 156 of file ng_hci.h.

7.49.1.295 #define NG_HCI_SCAN_REP_MODE1 0x01

Definition at line 157 of file ng_hci.h.

7.49.1.296 #define NG_HCI_SCAN_REP_MODE2 0x02

Definition at line 158 of file ng_hci.h.

7.49.1.297 #define NG_HCI_SCO_DATA_PKT 0x03

Definition at line 356 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_filter(), ng_h4_input(), ng_hci_drv_rcvdata(), ng_hci_sco_rcvdata(), ng_ubt_rcvdata(), and ubt_isoc_in_complete2().

7.49.1.298 #define NG_HCI_SCO_PKT_SIZE 0xff

Definition at line 357 of file ng_hci.h.

7.49.1.299 #define NG_HCI_SERVICE_TYPE_BEST_EFFORT 0x01

Definition at line 209 of file ng_hci.h.

Referenced by ng_btsocket_l2cap_attach(), and ng_l2cap_default_flow().

7.49.1.300 #define NG_HCI_SERVICE_TYPE_GUARANTEED 0x02

Definition at line 210 of file ng_hci.h.

7.49.1.301 #define NG_HCI_SERVICE_TYPE_NO_TRAFFIC 0x00

Definition at line 208 of file ng_hci.h.

7.49.1.302 #define NG_HCI_SPEC_V10 0x00

Definition at line 83 of file ng_hci.h.

7.49.1.303 #define NG_HCI_SPEC_V11 0x01

Definition at line 84 of file ng_hci.h.

7.49.1.304 #define NG_HCI_UNIT_COMMAND_PENDING (1 << 2)

Definition at line 486 of file ng_hci.h.

Referenced by complete_command(), con_compl(), ng_hci_command_timeout(), ng_hci_command_untimeout(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_process_command_timeout(), ng_hci_raw_rcvdata(), ng_hci_send_command(), and ng_hci_unit_clean().

7.49.1.305 #define NG_HCI_UNIT_CONNECTED (1 << 0)

Definition at line 483 of file ng_hci.h.

Referenced by ng_hci_connect(), and ng_hci_disconnect().

7.49.1.306 #define NG_HCI_UNIT_INITED (1 << 1)

Definition at line 484 of file ng_hci.h.

Referenced by ng_hci_default_rcvmsg(), ng_hci_disconnect(), and process_hc_baseband_params().

7.49.1.307 #define NG_HCI_UNIT_MODE_ACTIVE 0x00

Definition at line 142 of file ng_hci.h.

7.49.1.308 #define NG_HCI_UNIT_MODE_HOLD 0x01

Definition at line 143 of file ng_hci.h.

7.49.1.309 #define NG_HCI_UNIT_MODE_PARK 0x03

Definition at line 145 of file ng_hci.h.

7.49.1.310 #define NG_HCI_UNIT_MODE_SNIFF 0x02

Definition at line 144 of file ng_hci.h.

7.49.1.311 #define NG_HCI_UNIT_NAME_SIZE 248

Definition at line 80 of file ng_hci.h.

7.49.1.312 #define NG_HCI_UNIT_READY (NG_HCI_UNIT_CONNECTED|NG_HCI_UNIT_INITED)

Definition at line 485 of file ng_hci.h.

Referenced by ng_hci_drv_rcvdata(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_node_is_up(), process_info_params(), and send_data_packets().

7.49.1.313 #define NG_HCI_USE_SEMI_PERMANENT_LINK_KEYS 0x00

Definition at line 187 of file ng_hci.h.

7.49.1.314 #define NG_HCI_USE_TEMPORARY_LINK_KEY 0x01

Definition at line 188 of file ng_hci.h.

7.49.1.315 #define NG_HCI_WARN_LEVEL 3

Definition at line 479 of file ng_hci.h.

Referenced by ng_hci_constructor().

7.49.1.316 #define NG_HCI_XMIT_LEVEL_CURRENT 0x00

Definition at line 277 of file ng_hci.h.

7.49.1.317 #define NG_HCI_XMIT_LEVEL_MAXIMUM 0x01

Definition at line 278 of file ng_hci.h.

7.49.1.318 #define NGM_HCI_COOKIE 1000774184

Definition at line 58 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_hci_raw_send_ngmsg(), ng_btsocket_hci_raw_send_sync_ngmsg(), ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_ind(), ng_hci_lp_discon_ind(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_ind(), ng_hci_node_is_up(), ng_hci_upper_rcvmsg(), ng_l2cap_lower_rcvmsg(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_qos_req(), ng_l2cap_process_discon_timeout(), and sync_con_queue().

7.49.1.319 #define NGM_HCI_LP_CON_CFM 3

Definition at line 413 of file ng_hci.h.

Referenced by ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), and ng_l2cap_lower_rcvmsg().

7.49.1.320 #define NGM_HCI_LP_CON_IND 4

Definition at line 422 of file ng_hci.h.

Referenced by ng_hci_lp_con_ind(), and ng_l2cap_lower_rcvmsg().

7.49.1.321 #define NGM_HCI_LP_CON_REQ 1

Definition at line 394 of file ng_hci.h.

Referenced by ng_hci_upper_rcvmsg(), and ng_l2cap_lp_con_req().

7.49.1.322 #define NGM_HCI_LP_CON_RSP 5

Definition at line 430 of file ng_hci.h.

Referenced by ng_hci_upper_rcvmsg(), and ng_l2cap_lp_con_ind().

7.49.1.323 #define NGM_HCI_LP_DISCON_IND 6

Definition at line 438 of file ng_hci.h.

Referenced by ng_hci_lp_discon_ind(), and ng_l2cap_lower_rcvmsg().

7.49.1.324 #define NGM_HCI_LP_DISCON_REQ 2

Definition at line 406 of file ng_hci.h.

Referenced by ng_hci_upper_rcvmsg(), and ng_l2cap_process_discon_timeout().

7.49.1.325 #define NGM_HCI_LP_QOS_CFM 8

Definition at line 458 of file ng_hci.h.

Referenced by ng_hci_lp_qos_cfm(), and ng_l2cap_lower_rcvmsg().

7.49.1.326 #define NGM_HCI_LP_QOS_IND 9

Definition at line 465 of file ng_hci.h.

Referenced by ng_hci_lp_qos_ind(), and ng_l2cap_lower_rcvmsg().

7.49.1.327 #define NGM_HCI_LP_QOS_REQ 7

Definition at line 446 of file ng_hci.h.

Referenced by ng_hci_upper_rcvmsg(), and ng_l2cap_lp_qos_req().

7.49.1.328 #define NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE 109

Definition at line 542 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.329 #define NGM_HCI_NODE_GET_BDADDR 105

Definition at line 520 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.330 #define NGM_HCI_NODE_GET_BUFFER 104

Definition at line 508 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.331 #define NGM_HCI_NODE_GET_CON_LIST 111

Definition at line 560 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.332 #define NGM_HCI_NODE_GET_DEBUG 102

Definition at line 503 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.333 #define NGM_HCI_NODE_GET_FEATURES 106

Definition at line 524 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.334 #define NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK 114

Definition at line 595 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.335 #define NGM_HCI_NODE_GET_NEIGHBOR_CACHE 110

Definition at line 544 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.336 #define NGM_HCI_NODE_GET_PACKET_MASK 116

Definition at line 599 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.337 #define NGM_HCI_NODE_GET_ROLE_SWITCH 118

Definition at line 603 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.338 #define NGM_HCI_NODE_GET_STAT 107

Definition at line 527 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.339 #define NGM_HCI_NODE_GET_STATE 100

Definition at line 495 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.340 #define NGM_HCI_NODE_INIT 101

Definition at line 499 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.341 #define NGM_HCI_NODE_LIST_NAMES 200

Definition at line 607 of file ng_hci.h.

7.49.1.342 #define NGM_HCI_NODE_RESET_STAT 108

Definition at line 539 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.343 #define NGM_HCI_NODE_SET_DEBUG 103

Definition at line 504 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.344 #define NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK 115

Definition at line 596 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.345 #define NGM_HCI_NODE_SET_PACKET_MASK 117

Definition at line 600 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.346 #define NGM_HCI_NODE_SET_ROLE_SWITCH 119

Definition at line 604 of file ng_hci.h.

Referenced by ng_btsocket_hci_raw_control(), and ng_hci_default_rcvmsg().

7.49.1.347 #define NGM_HCI_NODE_UP 112

Definition at line 581 of file ng_hci.h.

Referenced by ng_hci_node_is_up(), and ng_l2cap_lower_rcvmsg().

7.49.1.348 #define NGM_HCI_SYNC_CON_QUEUE 113

Definition at line 589 of file ng_hci.h.

Referenced by ng_l2cap_lower_rcvmsg(), and sync_con_queue().

7.49.2 Typedef Documentation**7.49.2.1 typedef bdaddr_t* [bdaddr_p](#)**

Definition at line 377 of file ng_hci.h.

7.49.2.2 typedef ng_hci_status_rp [ng_hci_change_local_name_rp](#)

Definition at line 980 of file ng_hci.h.

7.49.2.3 typedef ng_hci_status_rp ng_hci_create_new_unit_key_rp

Definition at line 937 of file ng_hci.h.

7.49.2.4 typedef ng_hci_status_rp ng_hci_enable_unit_under_test_rp

Definition at line 1417 of file ng_hci.h.

7.49.2.5 typedef ng_hci_status_rp ng_hci_exit_periodic_inquiry_rp

Definition at line 642 of file ng_hci.h.

7.49.2.6 typedef ng_hci_status_rp ng_hci_h2hc_flow_control_rp

Definition at line 1200 of file ng_hci.h.

7.49.2.7 typedef ng_hci_status_rp ng_hci_host_buffer_size_rp

Definition at line 1210 of file ng_hci.h.

7.49.2.8 typedef ng_hci_status_rp ng_hci_inquiry_cancel_rp

Definition at line 627 of file ng_hci.h.

7.49.2.9 typedef u_int16_t ng_hci_node_debug_ep

Definition at line 505 of file ng_hci.h.

7.49.2.10 typedef u_int16_t ng_hci_node_link_policy_mask_ep

Definition at line 597 of file ng_hci.h.

7.49.2.11 typedef u_int16_t ng_hci_node_packet_mask_ep

Definition at line 601 of file ng_hci.h.

7.49.2.12 typedef u_int16_t ng_hci_node_role_switch_ep

Definition at line 605 of file ng_hci.h.

7.49.2.13 typedef u_int16_t ng_hci_node_state_ep

Definition at line 496 of file ng_hci.h.

7.49.2.14 [typedef ng_hci_status_rp ng_hci_periodic_inquiry_rp](#)

Definition at line 638 of file ng_hci.h.

7.49.2.15 [typedef ng_hci_status_rp ng_hci_reset_rp](#)

Definition at line 900 of file ng_hci.h.

7.49.2.16 [typedef ng_hci_status_rp ng_hci_set_event_filter_rp](#)

Definition at line 909 of file ng_hci.h.

7.49.2.17 [typedef ng_hci_status_rp ng_hci_set_event_mask_rp](#)

Definition at line 896 of file ng_hci.h.

7.49.2.18 [typedef ng_hci_status_rp ng_hci_write_auth_enable_rp](#)

Definition at line 1075 of file ng_hci.h.

7.49.2.19 [typedef ng_hci_status_rp ng_hci_write_con_accept_timo_rp](#)

Definition at line 1001 of file ng_hci.h.

7.49.2.20 [typedef ng_hci_status_rp ng_hci_write_encryption_mode_rp](#)

Definition at line 1089 of file ng_hci.h.

7.49.2.21 [typedef ng_hci_status_rp ng_hci_write_hold_mode_activity_rp](#)

Definition at line 1167 of file ng_hci.h.

7.49.2.22 [typedef ng_hci_status_rp ng_hci_write_iac_lap_rp](#)

Definition at line 1266 of file ng_hci.h.

7.49.2.23 [typedef ng_hci_status_rp ng_hci_write_inquiry_scan_activity_rp](#)

Definition at line 1061 of file ng_hci.h.

7.49.2.24 [typedef ng_hci_status_rp ng_hci_write_loopback_mode_rp](#)

Definition at line 1413 of file ng_hci.h.

7.49.2.25 typedef ng_hci_status_rp ng_hci_write_num_broadcast_retrans_rp

Definition at line 1153 of file ng_hci.h.

7.49.2.26 typedef ng_hci_status_rp ng_hci_write_page_scan_activity_rp

Definition at line 1045 of file ng_hci.h.

7.49.2.27 typedef ng_hci_status_rp ng_hci_write_page_scan_period_rp

Definition at line 1280 of file ng_hci.h.

7.49.2.28 typedef ng_hci_status_rp ng_hci_write_page_scan_rp

Definition at line 1294 of file ng_hci.h.

7.49.2.29 typedef ng_hci_status_rp ng_hci_write_page_timo_rp

Definition at line 1015 of file ng_hci.h.

7.49.2.30 typedef ng_hci_status_rp ng_hci_write_pin_type_rp

Definition at line 933 of file ng_hci.h.

7.49.2.31 typedef ng_hci_status_rp ng_hci_write_scan_enable_rp

Definition at line 1029 of file ng_hci.h.

7.49.2.32 typedef ng_hci_status_rp ng_hci_write_sco_flow_control_rp

Definition at line 1193 of file ng_hci.h.

7.49.2.33 typedef ng_hci_status_rp ng_hci_write_unit_class_rp

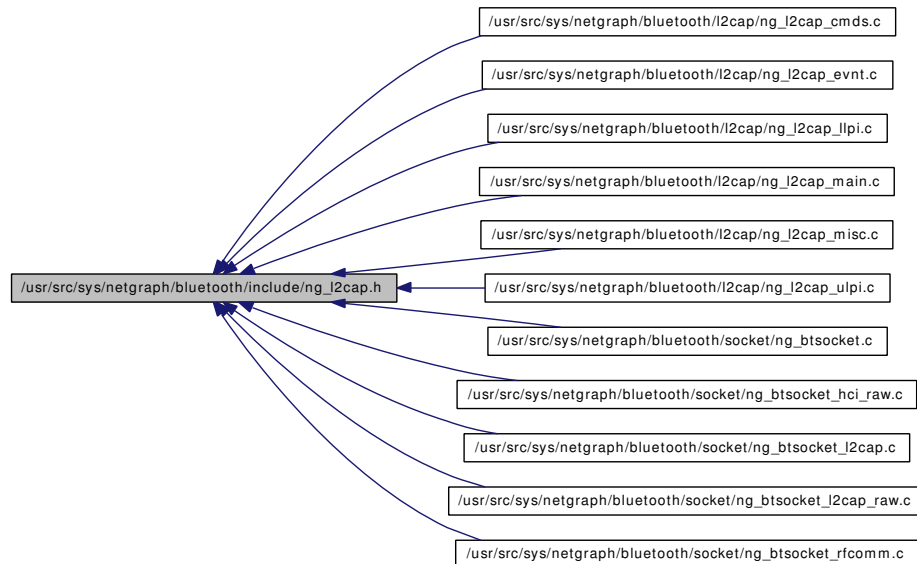
Definition at line 1103 of file ng_hci.h.

7.49.2.34 typedef ng_hci_status_rp ng_hci_write_voice_settings_rp

Definition at line 1117 of file ng_hci.h.

7.50 /usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- union [ng_l2cap_cmd_rej_data_t](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- union [ng_l2cap_cfg_opt_val_t](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)
- union [ng_l2cap_info_rsp_data_t](#)
- struct [__attribute__](#)
- struct [ng_l2cap_l2ca_con_ip](#)
- struct [ng_l2cap_l2ca_con_op](#)
- struct [ng_l2cap_l2ca_con_ind_ip](#)
- struct [ng_l2cap_l2ca_con_rsp_ip](#)
- struct [ng_l2cap_l2ca_con_rsp_op](#)
- struct [ng_l2cap_l2ca_cfg_ip](#)

- struct [ng_l2cap_l2ca_cfg_op](#)
- struct [ng_l2cap_l2ca_cfg_rsp_ip](#)
- struct [ng_l2cap_l2ca_cfg_rsp_op](#)
- struct [ng_l2cap_l2ca_cfg_ind_ip](#)
- struct [ng_l2cap_l2ca_qos_ind_ip](#)
- struct [ng_l2cap_l2ca_discon_ip](#)
- struct [ng_l2cap_l2ca_discon_op](#)
- struct [ng_l2cap_l2ca_write_op](#)
- struct [ng_l2cap_l2ca_grp_create_ip](#)
- struct [ng_l2cap_l2ca_grp_create_op](#)
- struct [ng_l2cap_l2ca_grp_close_ip](#)
- struct [ng_l2cap_l2ca_grp_add_member_ip](#)
- struct [ng_l2cap_l2ca_grp_add_member_op](#)
- struct [ng_l2cap_l2ca_grp_get_members_ip](#)
- struct [ng_l2cap_l2ca_grp_get_members_op](#)
- struct [ng_l2cap_l2ca_ping_ip](#)
- struct [ng_l2cap_l2ca_ping_op](#)
- struct [ng_l2cap_l2ca_get_info_ip](#)
- struct [ng_l2cap_l2ca_get_info_op](#)
- struct [ng_l2cap_l2ca_enable_clt_ip](#)
- struct [ng_l2cap_node_con_list_ep](#)
- struct [ng_l2cap_node_con_ep](#)
- struct [ng_l2cap_node_chan_list_ep](#)
- struct [ng_l2cap_node_chan_ep](#)

Defines

- #define [NG_L2CAP_HOOK_HCI](#) "hci"
- #define [NG_L2CAP_HOOK_L2C](#) "l2c"
- #define [NG_L2CAP_HOOK_CTL](#) "ctl"
- #define [NG_L2CAP_NODE_TYPE](#) "l2cap"
- #define [NGM_L2CAP_COOKIE](#) 1000774185
- #define [NG_L2CAP_NULL_CID](#) 0x0000
- #define [NG_L2CAP_SIGNAL_CID](#) 0x0001
- #define [NG_L2CAP_CLT_CID](#) 0x0002
- #define [NG_L2CAP_FIRST_CID](#) 0x0040
- #define [NG_L2CAP_LAST_CID](#) 0xffff
- #define [NG_L2CAP_MTU_MINIMUM](#) 48
- #define [NG_L2CAP_MTU_DEFAULT](#) 672
- #define [NG_L2CAP_MTU_MAXIMUM](#) 0xffff
- #define [NG_L2CAP_FLUSH_TIMO_DEFAULT](#) 0xffff
- #define [NG_L2CAP_LINK_TIMO_DEFAULT](#) 0xffff
- #define [NG_L2CAP_REJ_NOT_UNDERSTOOD](#) 0x0000
- #define [NG_L2CAP_REJ_MTU_EXCEEDED](#) 0x0001
- #define [NG_L2CAP_REJ_INVALID_CID](#) 0x0002
- #define [NG_L2CAP_PSM_ANY](#) 0x0000
- #define [NG_L2CAP_PSM_SDP](#) 0x0001
- #define [NG_L2CAP_PSM_RFCOMM](#) 0x0003
- #define [NG_L2CAP_PSM_TCP](#) 0x0005
- #define [NG_L2CAP_SUCCESS](#) 0x0000

- #define NG_L2CAP_PENDING 0x0001
- #define NG_L2CAP_PSM_NOT_SUPPORTED 0x0002
- #define NG_L2CAP_SEQUIRY_BLOCK 0x0003
- #define NG_L2CAP_NO_RESOURCES 0x0004
- #define NG_L2CAP_TIMEOUT 0xeeee
- #define NG_L2CAP_UNKNOWN 0xffff
- #define NG_L2CAP_NO_INFO 0x0000
- #define NG_L2CAP_AUTH_PENDING 0x0001
- #define NG_L2CAP_AUTZ_PENDING 0x0002
- #define NG_L2CAP_UNACCEPTABLE_PARAMS 0x0001
- #define NG_L2CAP_REJECT 0x0002
- #define NG_L2CAP_UNKNOWN_OPTION 0x0003
- #define NG_L2CAP_OPT_CFLAG_BIT 0x0001
- #define NG_L2CAP_OPT_CFLAG(flags) ((flags) & NG_L2CAP_OPT_CFLAG_BIT)
- #define NG_L2CAP_OPT_HINT_BIT 0x80
- #define NG_L2CAP_OPT_HINT(type) ((type) & NG_L2CAP_OPT_HINT_BIT)
- #define NG_L2CAP_OPT_HINT_MASK 0x7f
- #define NG_L2CAP_OPT_MTU 0x01
- #define NG_L2CAP_OPT_MTU_SIZE sizeof(u_int16_t)
- #define NG_L2CAP_OPT_FLUSH_TIMO 0x02
- #define NG_L2CAP_OPT_FLUSH_TIMO_SIZE sizeof(u_int16_t)
- #define NG_L2CAP_OPT_QOS 0x03
- #define NG_L2CAP_OPT_QOS_SIZE sizeof(ng_l2cap_flow_t)
- #define NG_L2CAP_CONNLESS_MTU 0x0001
- #define NG_L2CAP_NOT_SUPPORTED 0x0001
- #define NG_L2CAP_CLT_MTU_MAXIMUM (NG_L2CAP_MTU_MAXIMUM - sizeof(ng_l2cap_clt_hdr_t))
- #define NG_L2CAP_CMD_REJ 0x01
- #define NG_L2CAP_CON_REQ 0x02
- #define NG_L2CAP_CON_RSP 0x03
- #define NG_L2CAP_CFG_REQ 0x04
- #define NG_L2CAP_CFG_RSP 0x05
- #define NG_L2CAP_DISCON_REQ 0x06
- #define NG_L2CAP_DISCON_RSP 0x07
- #define NG_L2CAP_ECHO_REQ 0x08
- #define NG_L2CAP_ECHO_RSP 0x09
- #define NG_L2CAP_MAX_ECHO_SIZE (NG_L2CAP_MTU_MAXIMUM - sizeof(ng_l2cap_cmd_hdr_t))
- #define NG_L2CAP_INFO_REQ 0x0a
- #define NG_L2CAP_INFO_RSP 0x0b
- #define NGM_L2CAP_L2CA_CON 0x80
- #define NGM_L2CAP_L2CA_CON_IND 0x81
- #define NGM_L2CAP_L2CA_CON_RSP 0x82
- #define NGM_L2CAP_L2CA_CFG 0x83
- #define NGM_L2CAP_L2CA_CFG_RSP 0x84
- #define NGM_L2CAP_L2CA_CFG_IND 0x85
- #define NGM_L2CAP_L2CA_QOS_IND 0x86
- #define NGM_L2CAP_L2CA_DISCON 0x87
- #define NGM_L2CAP_L2CA_DISCON_IND 0x88
- #define NGM_L2CAP_L2CA_WRITE 0x89

- #define `NGM_L2CAP_L2CA_GRP_CREATE` 0x8a
- #define `NGM_L2CAP_L2CA_GRP_CLOSE` 0x8b
- #define `NGM_L2CAP_L2CA_GRP_ADD_MEMBER` 0x8c
- #define `NGM_L2CAP_L2CA_GRP_REM_MEMBER` 0x8d
- #define `NGM_L2CAP_L2CA_GRP_MEMBERSHIP` 0x8e
- #define `NGM_L2CAP_L2CA_PING` 0x8f
- #define `NGM_L2CAP_L2CA_GET_INFO` 0x90
- #define `NGM_L2CAP_L2CA_ENABLE_CLT` 0x91
- #define `NG_L2CAP_CON_CLOSED` 0
- #define `NG_L2CAP_W4_LP_CON_CFM` 1
- #define `NG_L2CAP_CON_OPEN` 2
- #define `NG_L2CAP_CLOSED` 0
- #define `NG_L2CAP_W4_L2CAP_CON_RSP` 1
- #define `NG_L2CAP_W4_L2CA_CON_RSP` 2
- #define `NG_L2CAP_CONFIG` 3
- #define `NG_L2CAP_OPEN` 4
- #define `NG_L2CAP_W4_L2CAP_DISCON_RSP` 5
- #define `NG_L2CAP_W4_L2CA_DISCON_RSP` 6
- #define `NG_L2CAP_CLT_SDP_DISABLED` (1 << 0)
- #define `NG_L2CAP_CLT_RFCOMM_DISABLED` (1 << 1)
- #define `NG_L2CAP_CLT_TCP_DISABLED` (1 << 2)
- #define `NG_L2CAP_ALERT_LEVEL` 1
- #define `NG_L2CAP_ERR_LEVEL` 2
- #define `NG_L2CAP_WARN_LEVEL` 3
- #define `NG_L2CAP_INFO_LEVEL` 4
- #define `NGM_L2CAP_NODE_GET_FLAGS` 0x400
- #define `NGM_L2CAP_NODE_GET_DEBUG` 0x401
- #define `NGM_L2CAP_NODE_SET_DEBUG` 0x402
- #define `NGM_L2CAP_NODE_HOOK_INFO` 0x409
- #define `NGM_L2CAP_NODE_GET_CON_LIST` 0x40a
- #define `NG_L2CAP_CON_TX` (1 << 0)
- #define `NG_L2CAP_CON_RX` (1 << 1)
- #define `NG_L2CAP_CON_OUTGOING` (1 << 2)
- #define `NG_L2CAP_CON_LP_TIMO` (1 << 3)
- #define `NG_L2CAP_CON_AUTO_DISCON_TIMO` (1 << 4)
- #define `NG_L2CAP_CON_DYING` (1 << 5)
- #define `NG_L2CAP_MAX_CON_NUM` ((0xffff - sizeof(ng_l2cap_node_con_list_ep))/sizeof(ng_l2cap_node_con_ep))
- #define `NGM_L2CAP_NODE_GET_CHAN_LIST` 0x40b
- #define `NG_L2CAP_MAX_CHAN_NUM` ((0xffff - sizeof(ng_l2cap_node_chan_list_ep))/sizeof(ng_l2cap_node_chan_ep))
- #define `NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO` 0x40c
- #define `NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO` 0x40d

Typedefs

- typedef ng_l2cap_flow_t * ng_l2cap_flow_p
- typedef ng_l2cap_cmd_rej_data_t * ng_l2cap_cmd_rej_data_p
- typedef ng_l2cap_cfg_opt_t * ng_l2cap_cfg_opt_p
- typedef ng_l2cap_cfg_opt_val_t * ng_l2cap_cfg_opt_val_p
- typedef ng_l2cap_discon_req_cp ng_l2cap_discon_rsp_cp
- typedef ng_l2cap_info_rsp_data_t * ng_l2cap_info_rsp_data_p
- typedef ng_l2cap_l2ca_discon_ip ng_l2cap_l2ca_discon_ind_ip
- typedef ng_l2cap_l2ca_grp_add_member_ip ng_l2cap_l2ca_grp_rem_member_ip
- typedef u_int16_t ng_l2cap_node_flags_ep
- typedef u_int16_t ng_l2cap_node_debug_ep
- typedef u_int16_t ng_l2cap_node_auto_discon_ep

7.50.1 Define Documentation

7.50.1.1 #define NG_L2CAP_ALERT_LEVEL 1

Definition at line 598 of file ng_l2cap.h.

7.50.1.2 #define NG_L2CAP_AUTH_PENDING 0x0001

Definition at line 114 of file ng_l2cap.h.

7.50.1.3 #define NG_L2CAP_AUTZ_PENDING 0x0002

Definition at line 115 of file ng_l2cap.h.

7.50.1.4 #define NG_L2CAP_CFG_REQ 0x04

Definition at line 223 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_signal_cmd().

7.50.1.5 #define NG_L2CAP_CFG_RSP 0x05

Definition at line 231 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_process_signal_cmd(), and send_l2cap_cfg_rsp().

7.50.1.6 #define NG_L2CAP_CLOSED 0

Definition at line 584 of file ng_l2cap.h.

Referenced by ng_l2cap_new_chan().

7.50.1.7 #define NG_L2CAP_CLT_CID 0x0002

Definition at line 75 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_data_input(), ng_l2cap_con_wakeup(), and ng_l2cap_receive().

7.50.1.8 #define NG_L2CAP_CLT_MTU_MAXIMUM (NG_L2CAP_MTU_MAXIMUM - sizeof(ng_l2cap_clt_hdr_t))

Definition at line 175 of file ng_l2cap.h.

7.50.1.9 #define NG_L2CAP_CLT_RFCOMM_DISABLED (1 << 1)

Definition at line 594 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_clt_receive(), and ng_l2cap_l2ca_enable_clt().

7.50.1.10 #define NG_L2CAP_CLT_SDP_DISABLED (1 << 0)

Definition at line 593 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_clt_receive(), and ng_l2cap_l2ca_enable_clt().

7.50.1.11 #define NG_L2CAP_CLT_TCP_DISABLED (1 << 2)

Definition at line 595 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_clt_receive(), and ng_l2cap_l2ca_enable_clt().

7.50.1.12 #define NG_L2CAP_CMD_REJ 0x01

Definition at line 186 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_signal_cmd(), and send_l2cap_reject().

7.50.1.13 #define NG_L2CAP_CON_AUTO_DISCON_TIMO (1 << 4)

Definition at line 625 of file ng_l2cap.h.

Referenced by ng_l2cap_cleanup(), ng_l2cap_con_ref(), ng_l2cap_discon_timeout(), ng_l2cap_discon_untimout(), ng_l2cap_free_con(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_timeout(), and ng_l2cap_process_discon_timeout().

7.50.1.14 #define NG_L2CAP_CON_CLOSED 0

Definition at line 579 of file ng_l2cap.h.

Referenced by ng_l2cap_free_con(), and ng_l2cap_new_con().

7.50.1.15 #define NG_L2CAP_CON_DYING (1 << 5)

Definition at line 626 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), and ng_l2cap_con_unref().

7.50.1.16 #define NG_L2CAP_CON_LP_TIMO (1 << 3)

Definition at line 624 of file ng_l2cap.h.

Referenced by ng_l2cap_cleanup(), ng_l2cap_discon_timeout(), ng_l2cap_free_con(), ng_l2cap_lp_timeout(), ng_l2cap_lp_untimeout(), and ng_l2cap_process_lp_timeout().

7.50.1.17 #define NG_L2CAP_CON_OPEN 2

Definition at line 581 of file ng_l2cap.h.

Referenced by ng_l2cap_con_ref(), ng_l2cap_con_unref(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_deliver(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), and ng_l2cap_lp_receive().

7.50.1.18 #define NG_L2CAP_CON_OUTGOING (1 << 2)

Definition at line 623 of file ng_l2cap.h.

Referenced by ng_l2cap_con_ref(), ng_l2cap_con_unref(), and ng_l2cap_lp_con_req().

7.50.1.19 #define NG_L2CAP_CON_REQ 0x02

Definition at line 207 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_con_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_signal_cmd().

7.50.1.20 #define NG_L2CAP_CON_RSP 0x03

Definition at line 214 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_process_signal_cmd(), and send_l2cap_con_rej().

7.50.1.21 #define NG_L2CAP_CON_RX (1 << 1)

Definition at line 622 of file ng_l2cap.h.

Referenced by ng_l2cap_default_rcvmsg().

7.50.1.22 #define NG_L2CAP_CON_TX (1 << 0)

Definition at line 621 of file ng_l2cap.h.

Referenced by ng_l2cap_default_rcvmsg().

7.50.1.23 #define NG_L2CAP_CONFIG 3

Definition at line 587 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_con_rsp(), and ng_l2cap_process_discon_req().

7.50.1.24 #define NG_L2CAP_CONNLESS_MTU 0x0001

Definition at line 139 of file ng_l2cap.h.

Referenced by ng_l2cap_process_info_req(), and ng_l2cap_process_info_rsp().

7.50.1.25 #define NG_L2CAP_DISCON_REQ 0x06

Definition at line 256 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_discon_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_signal_cmd().

7.50.1.26 #define NG_L2CAP_DISCON_RSP 0x07

Definition at line 263 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_discon_req(), and ng_l2cap_process_signal_cmd().

7.50.1.27 #define NG_L2CAP_ECHO_REQ 0x08

Definition at line 267 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_ping_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_signal_cmd().

7.50.1.28 #define NG_L2CAP_ECHO_RSP 0x09

Definition at line 271 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_echo_req(), and ng_l2cap_process_signal_cmd().

7.50.1.29 #define NG_L2CAP_ERR_LEVEL 2

Definition at line 599 of file ng_l2cap.h.

7.50.1.30 #define NG_L2CAP_FIRST_CID 0x0040

Definition at line 77 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_data_input(), ng_l2cap_get_cid(), and ng_l2cap_l2ca_write_req().

7.50.1.31 #define NG_L2CAP_FLUSH_TIMO_DEFAULT 0xffff

Definition at line 86 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_attach(), ng_l2cap_l2ca_cfg_req(), and ng_l2cap_new_chan().

7.50.1.32 #define NG_L2CAP_HOOK_CTL "ctl"

Definition at line 54 of file ng_l2cap.h.

Referenced by ng_l2cap_default_rcvmsg(), and ng_l2cap_newhook().

7.50.1.33 #define NG_L2CAP_HOOK_HCI "hci"

Definition at line 52 of file ng_l2cap.h.

Referenced by ng_l2cap_default_rcvmsg(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_lp_qos_req(), ng_l2cap_newhook(), and ng_l2cap_process_discon_timeout().

7.50.1.34 #define NG_L2CAP_HOOK_L2C "l2c"

Definition at line 53 of file ng_l2cap.h.

Referenced by ng_l2cap_default_rcvmsg(), and ng_l2cap_newhook().

7.50.1.35 #define NG_L2CAP_INFO_LEVEL 4

Definition at line 601 of file ng_l2cap.h.

7.50.1.36 #define NG_L2CAP_INFO_REQ 0x0a

Definition at line 277 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_signal_cmd().

7.50.1.37 #define NG_L2CAP_INFO_RSP 0x0b

Definition at line 283 of file ng_l2cap.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_info_req(), and ng_l2cap_process_signal_cmd().

7.50.1.38 #define NG_L2CAP_LAST_CID 0xffff

Definition at line 78 of file ng_l2cap.h.

7.50.1.39 #define NG_L2CAP_LINK_TIMO_DEFAULT 0xffff

Definition at line 87 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_attach(), and ng_l2cap_new_chan().

7.50.1.40 #define NG_L2CAP_MAX_CHAN_NUM ((0xffff - sizeof(ng_l2cap_node_chan_list - ep))/sizeof(ng_l2cap_node_chan_ep))

Definition at line 657 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_default_rcvmsg().

7.50.1.41 #define NG_L2CAP_MAX_CON_NUM ((0xffff - sizeof(ng_l2cap_node_con_list - ep))/sizeof(ng_l2cap_node_con_ep))

Definition at line 636 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_default_rcvmsg().

7.50.1.42 #define NG_L2CAP_MAX_ECHO_SIZE (NG_L2CAP_MTU_MAXIMUM - sizeof(ng_l2cap_cmd_hdr_t))

Definition at line 272 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_l2ca_ping_req().

7.50.1.43 #define NG_L2CAP_MTU_DEFAULT 672

Definition at line 82 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_attach(), ng_btsocket_l2cap_data_input(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_new_chan(), and ng_l2cap_process_info_req().

7.50.1.44 #define NG_L2CAP_MTU_MAXIMUM 0xffff

Definition at line 83 of file ng_l2cap.h.

7.50.1.45 #define NG_L2CAP_MTU_MINIMUM 48

Definition at line 81 of file ng_l2cap.h.

7.50.1.46 #define NG_L2CAP_NO_INFO 0x0000

Definition at line 113 of file ng_l2cap.h.

7.50.1.47 #define NG_L2CAP_NO_RESOURCES 0x0004

Definition at line 107 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_con_ind(), ng_l2cap_con_wakeup(), and ng_l2cap_process_con_req().

7.50.1.48 #define NG_L2CAP_NODE_TYPE "l2cap"

Definition at line 57 of file ng_l2cap.h.

7.50.1.49 #define NG_L2CAP_NOT_SUPPORTED 0x0001

Definition at line 143 of file ng_l2cap.h.

Referenced by ng_l2cap_process_info_req().

7.50.1.50 #define NG_L2CAP_NULL_CID 0x0000

Definition at line 73 of file ng_l2cap.h.

Referenced by ng_l2cap_get_cid(), and ng_l2cap_new_chan().

7.50.1.51 #define NG_L2CAP_OPEN 4

Definition at line 588 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_req(), ng_l2cap_process_cfg_req(), and ng_l2cap_process_discon_req().

7.50.1.52 #define NG_L2CAP_OPT_CFLAG(flags) ((flags) & NG_L2CAP_OPT_CFLAG_BIT)

Definition at line 126 of file ng_l2cap.h.

Referenced by ng_l2cap_process_cfg_req(), and ng_l2cap_process_cfg_rsp().

7.50.1.53 #define NG_L2CAP_OPT_CFLAG_BIT 0x0001

Definition at line 125 of file ng_l2cap.h.

7.50.1.54 #define NG_L2CAP_OPT_FLUSH_TIMO 0x02

Definition at line 132 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt(), ng_l2cap_process_cfg_req(), and ng_l2cap_process_cfg_rsp().

7.50.1.55 #define NG_L2CAP_OPT_FLUSH_TIMO_SIZE sizeof(u_int16_t)

Definition at line 133 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt().

7.50.1.56 #define NG_L2CAP_OPT_HINT(type) ((type) & NG_L2CAP_OPT_HINT_BIT)

Definition at line 128 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt().

7.50.1.57 #define NG_L2CAP_OPT_HINT_BIT 0x80

Definition at line 127 of file ng_l2cap.h.

7.50.1.58 #define NG_L2CAP_OPT_HINT_MASK 0x7f

Definition at line 129 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt().

7.50.1.59 #define NG_L2CAP_OPT_MTU 0x01

Definition at line 130 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt(), ng_l2cap_process_cfg_req(), and ng_l2cap_process_cfg_rsp().

7.50.1.60 #define NG_L2CAP_OPT_MTU_SIZE sizeof(u_int16_t)

Definition at line 131 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt().

7.50.1.61 #define NG_L2CAP_OPT_QOS 0x03

Definition at line 134 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt(), ng_l2cap_process_cfg_req(), and ng_l2cap_process_cfg_rsp().

7.50.1.62 #define NG_L2CAP_OPT_QOS_SIZE sizeof(ng_l2cap_flow_t)

Definition at line 135 of file ng_l2cap.h.

Referenced by get_next_l2cap_opt().

7.50.1.63 #define NG_L2CAP_PENDING 0x0001

Definition at line 104 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_l2cap_l2ca_con_rsp_req(), and ng_l2cap_process_con_rsp().

7.50.1.64 #define NG_L2CAP_PSM_ANY 0x0000

Definition at line 96 of file ng_l2cap.h.

7.50.1.65 #define NG_L2CAP_PSM_NOT_SUPPORTED 0x0002

Definition at line 105 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_process_l2ca_con_ind(), ng_l2cap_l2ca_enable_clt(), and ng_l2cap_process_con_req().

7.50.1.66 #define NG_L2CAP_PSM_RFCOMM 0x0003

Definition at line 98 of file ng_l2cap.h.

Referenced by `ng_btsocket_rfcomm_session_create()`, `ng_l2cap_l2ca_clt_receive()`, and `ng_l2cap_l2ca_enable_clt()`.

7.50.1.67 #define NG_L2CAP_PSM_SDP 0x0001

Definition at line 97 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_l2ca_clt_receive()`, and `ng_l2cap_l2ca_enable_clt()`.

7.50.1.68 #define NG_L2CAP_PSM_TCP 0x0005

Definition at line 99 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_l2ca_clt_receive()`, and `ng_l2cap_l2ca_enable_clt()`.

7.50.1.69 #define NG_L2CAP_REJ_INVALID_CID 0x0002

Definition at line 92 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_con_rsp()`, and `ng_l2cap_process_discon_req()`.

7.50.1.70 #define NG_L2CAP_REJ_MTU_EXCEEDED 0x0001

Definition at line 91 of file `ng_l2cap.h`.

7.50.1.71 #define NG_L2CAP_REJ_NOT_UNDERSTOOD 0x0000

Definition at line 90 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_signal_cmd()`.

7.50.1.72 #define NG_L2CAP_REJECT 0x0002

Definition at line 120 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_cfg_req()`.

7.50.1.73 #define NG_L2CAP_SEQUIRY_BLOCK 0x0003

Definition at line 106 of file `ng_l2cap.h`.

7.50.1.74 #define NG_L2CAP_SIGNAL_CID 0x0001

Definition at line 74 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_con_wakeup()`, and `ng_l2cap_receive()`.

7.50.1.75 #define NG_L2CAP_SUCCESS 0x0000

Definition at line 103 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_rsp()`, `ng_l2cap_process_echo_rsp()`, `ng_l2cap_process_info_req()`, and `ng_l2cap_process_info_rsp()`.

7.50.1.76 **#define NG_L2CAP_TIMEOUT 0xeeee**

Definition at line 108 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_lp_timeout()`.

7.50.1.77 **#define NG_L2CAP_UNACCEPTABLE_PARAMS 0x0001**

Definition at line 119 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`.

7.50.1.78 **#define NG_L2CAP_UNKNOWN 0xffff**

Definition at line 109 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_process_cfg_rsp()`, and `ng_l2cap_process_info_rsp()`.

7.50.1.79 **#define NG_L2CAP_UNKNOWN_OPTION 0x0003**

Definition at line 121 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, and `ng_l2cap_process_cfg_req()`.

7.50.1.80 **#define NG_L2CAP_W4_L2CA_CON_RSP 2**

Definition at line 586 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_l2ca_con_rsp_req()`, and `ng_l2cap_process_con_req()`.

7.50.1.81 **#define NG_L2CAP_W4_L2CA_DISCON_RSP 6**

Definition at line 590 of file `ng_l2cap.h`.

7.50.1.82 **#define NG_L2CAP_W4_L2CAP_CON_RSP 1**

Definition at line 585 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_l2ca_con_req()`, and `ng_l2cap_process_con_rsp()`.

7.50.1.83 **#define NG_L2CAP_W4_L2CAP_DISCON_RSP 5**

Definition at line 589 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_process_discon_req()`, and `ng_l2cap_process_discon_rsp()`.

7.50.1.84 #define NG_L2CAP_W4_LP_CON_CFM 1

Definition at line 580 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

7.50.1.85 #define NG_L2CAP_WARN_LEVEL 3

Definition at line 600 of file `ng_l2cap.h`.

Referenced by `ng_l2cap_constructor()`.

7.50.1.86 #define NGM_L2CAP_COOKIE 1000774185

Definition at line 58 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_node_rcvmsg()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_send_ngmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_send_hook_info()`, and `ng_l2cap_upper_rcvmsg()`.

7.50.1.87 #define NGM_L2CAP_L2CA_CFG 0x83

Definition at line 382 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_l2ca_msg_input()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp()`, and `ng_l2cap_upper_rcvmsg()`.

7.50.1.88 #define NGM_L2CAP_L2CA_CFG_IND 0x85

Definition at line 415 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_l2ca_msg_input()`, and `ng_l2cap_l2ca_cfg_ind()`.

7.50.1.89 #define NGM_L2CAP_L2CA_CFG_RSP 0x84

Definition at line 401 of file `ng_l2cap.h`.

Referenced by `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_l2ca_msg_input()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, and `ng_l2cap_upper_rcvmsg()`.

7.50.1.90 #define NGM_L2CAP_L2CA_CON 0x80

Definition at line 338 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_send_l2ca_con_req(), ng_l2cap_l2ca_con_rsp(), and ng_l2cap_upper_rcvmsg().

7.50.1.91 #define NGM_L2CAP_L2CA_CON_IND 0x81

Definition at line 353 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), and ng_l2cap_l2ca_con_ind().

7.50.1.92 #define NGM_L2CAP_L2CA_CON_RSP 0x82

Definition at line 365 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_send_l2ca_con_rsp_req(), ng_l2cap_l2ca_con_rsp_rsp(), and ng_l2cap_upper_rcvmsg().

7.50.1.93 #define NGM_L2CAP_L2CA_DISCON 0x87

Definition at line 434 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_l2cap_l2ca_discon_rsp(), and ng_l2cap_upper_rcvmsg().

7.50.1.94 #define NGM_L2CAP_L2CA_DISCON_IND 0x88

Definition at line 446 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), and ng_l2cap_l2ca_discon_ind().

7.50.1.95 #define NGM_L2CAP_L2CA_ENABLE_CLT 0x91

Definition at line 558 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_enable_clt(), and ng_l2cap_upper_rcvmsg().

7.50.1.96 #define NGM_L2CAP_L2CA_GET_INFO 0x90

Definition at line 543 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), ng_l2cap_l2ca_get_info_rsp(), and ng_l2cap_upper_rcvmsg().

7.50.1.97 #define NGM_L2CAP_L2CA_GRP_ADD_MEMBER 0x8c

Definition at line 489 of file ng_l2cap.h.

Referenced by ng_l2cap_upper_rcvmsg().

7.50.1.98 #define NGM_L2CAP_L2CA_GRP_CLOSE 0x8b

Definition at line 475 of file ng_l2cap.h.

Referenced by ng_l2cap_upper_rcvmsg().

7.50.1.99 #define NGM_L2CAP_L2CA_GRP_CREATE 0x8a

Definition at line 463 of file ng_l2cap.h.

Referenced by ng_l2cap_upper_rcvmsg().

7.50.1.100 #define NGM_L2CAP_L2CA_GRP_MEMBERSHIP 0x8e

Definition at line 512 of file ng_l2cap.h.

Referenced by ng_l2cap_upper_rcvmsg().

7.50.1.101 #define NGM_L2CAP_L2CA_GRP_REM_MEMBER 0x8d

Definition at line 502 of file ng_l2cap.h.

Referenced by ng_l2cap_upper_rcvmsg().

7.50.1.102 #define NGM_L2CAP_L2CA_PING 0x8f

Definition at line 526 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), ng_l2cap_l2ca_ping_rsp(), and ng_l2cap_upper_rcvmsg().

7.50.1.103 #define NGM_L2CAP_L2CA_QOS_IND 0x86

Definition at line 426 of file ng_l2cap.h.

Referenced by ng_l2cap_l2ca_qos_ind().

7.50.1.104 #define NGM_L2CAP_L2CA_WRITE 0x89

Definition at line 452 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_l2ca_write_req(), and ng_l2cap_l2ca_write_rsp().

7.50.1.105 #define NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO 0x40c

Definition at line 660 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), and ng_l2cap_default_rcvmsg().

7.50.1.106 #define NGM_L2CAP_NODE_GET_CHAN_LIST 0x40b

Definition at line 639 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), and ng_l2cap_default_rcvmsg().

7.50.1.107 #define NGM_L2CAP_NODE_GET_CON_LIST 0x40a

Definition at line 615 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), and ng_l2cap_default_rcvmsg().

7.50.1.108 #define NGM_L2CAP_NODE_GET_DEBUG 0x401

Definition at line 608 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), and ng_l2cap_default_rcvmsg().

7.50.1.109 #define NGM_L2CAP_NODE_GET_FLAGS 0x400

Definition at line 604 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), and ng_l2cap_default_rcvmsg().

7.50.1.110 #define NGM_L2CAP_NODE_HOOK_INFO 0x409

Definition at line 612 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_node_rcvmsg(), and ng_l2cap_send_hook_info().

7.50.1.111 #define NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO 0x40d

Definition at line 661 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_default_rcvmsg().

7.50.1.112 #define NGM_L2CAP_NODE_SET_DEBUG 0x402

Definition at line 609 of file ng_l2cap.h.

Referenced by ng_btsocket_l2cap_raw_control(), and ng_l2cap_default_rcvmsg().

7.50.2 Typedef Documentation**7.50.2.1 typedef ng_l2cap_cfg_opt_t* ng_l2cap_cfg_opt_p**

Definition at line 245 of file ng_l2cap.h.

7.50.2.2 typedef [ng_l2cap_cfg_opt_val_t](#)* [ng_l2cap_cfg_opt_val_p](#)

Definition at line 253 of file [ng_l2cap.h](#).

7.50.2.3 typedef [ng_l2cap_cmd_rej_data_t](#)* [ng_l2cap_cmd_rej_data_p](#)

Definition at line 204 of file [ng_l2cap.h](#).

7.50.2.4 typedef [ng_l2cap_discon_req_cp](#) [ng_l2cap_discon_rsp_cp](#)

Definition at line 264 of file [ng_l2cap.h](#).

7.50.2.5 typedef [ng_l2cap_flow_t](#)* [ng_l2cap_flow_p](#)

Definition at line 156 of file [ng_l2cap.h](#).

7.50.2.6 typedef [ng_l2cap_info_rsp_data_t](#)* [ng_l2cap_info_rsp_data_p](#)

Definition at line 299 of file [ng_l2cap.h](#).

7.50.2.7 typedef [ng_l2cap_l2ca_discon_ip](#) [ng_l2cap_l2ca_discon_ind_ip](#)

Definition at line 448 of file [ng_l2cap.h](#).

7.50.2.8 typedef [ng_l2cap_l2ca_grp_add_member_ip](#) [ng_l2cap_l2ca_grp_rem_member_ip](#)

Definition at line 504 of file [ng_l2cap.h](#).

7.50.2.9 typedef [u_int16_t](#) [ng_l2cap_node_auto_discon_ep](#)

Definition at line 662 of file [ng_l2cap.h](#).

7.50.2.10 typedef [u_int16_t](#) [ng_l2cap_node_debug_ep](#)

Definition at line 610 of file [ng_l2cap.h](#).

7.50.2.11 typedef [u_int16_t](#) [ng_l2cap_node_flags_ep](#)

Definition at line 605 of file [ng_l2cap.h](#).

7.51 /usr/src/sys/netgraph/bluetooth/include/ng_ubt.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_ubt_node_qlen_ep](#)
- struct [ng_ubt_node_stat_ep](#)

Defines

- #define [NG_UBT_NODE_TYPE](#) "ubt"
- #define [NG_UBT_HOOK](#) "hook"
- #define [NGM_UBT_COOKIE](#) 1021837971
- #define [NG_UBT_ALERT_LEVEL](#) 1
- #define [NG_UBT_ERR_LEVEL](#) 2
- #define [NG_UBT_WARN_LEVEL](#) 3
- #define [NG_UBT_INFO_LEVEL](#) 4
- #define [NGM_UBT_NODE_SET_DEBUG](#) 1
- #define [NGM_UBT_NODE_GET_DEBUG](#) 2
- #define [NGM_UBT_NODE_SET_QLEN](#) 3
- #define [NGM_UBT_NODE_GET_QLEN](#) 4
- #define [NGM_UBT_NODE_QUEUE_CMD](#) 1
- #define [NGM_UBT_NODE_QUEUE_ACL](#) 2
- #define [NGM_UBT_NODE_QUEUE_SCO](#) 3
- #define [NGM_UBT_NODE_GET_STAT](#) 5
- #define [NGM_UBT_NODE_RESET_STAT](#) 6
- #define [NGM_UBT_NODE_DEV_NODES](#) 7

Typedefs

- typedef u_int16_t [ng_ubt_node_debug_ep](#)
- typedef u_int16_t [ng_ubt_node_dev_nodes_ep](#)

7.51.1 Define Documentation

7.51.1.1 #define NG_UBT_ALERT_LEVEL 1

Definition at line 49 of file [ng_ubt.h](#).

7.51.1.2 #define NG_UBT_ERR_LEVEL 2

Definition at line 50 of file [ng_ubt.h](#).

7.51.1.3 #define NG_UBT_HOOK "hook"

Definition at line 44 of file ng_ubt.h.

Referenced by ng_ubt_newhook(), and ng_ubt_rcvmsg().

7.51.1.4 #define NG_UBT_INFO_LEVEL 4

Definition at line 52 of file ng_ubt.h.

7.51.1.5 #define NG_UBT_NODE_TYPE "ubt"

Definition at line 43 of file ng_ubt.h.

Referenced by ubt_modevent().

7.51.1.6 #define NG_UBT_WARN_LEVEL 3

Definition at line 51 of file ng_ubt.h.

Referenced by USB_ATTACH().

7.51.1.7 #define NGM_UBT_COOKIE 1021837971

Definition at line 46 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.8 #define NGM_UBT_NODE_DEV_NODES 7

Definition at line 87 of file ng_ubt.h.

7.51.1.9 #define NGM_UBT_NODE_GET_DEBUG 2

Definition at line 61 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.10 #define NGM_UBT_NODE_GET_QLEN 4

Definition at line 65 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.11 #define NGM_UBT_NODE_GET_STAT 5

Definition at line 75 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.12 #define NGM_UBT_NODE_QUEUE_ACL 2

Definition at line 69 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.13 #define NGM_UBT_NODE_QUEUE_CMD 1

Definition at line 68 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.14 #define NGM_UBT_NODE_QUEUE_SCO 3

Definition at line 70 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.15 #define NGM_UBT_NODE_RESET_STAT 6

Definition at line 85 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.16 #define NGM_UBT_NODE_SET_DEBUG 1

Definition at line 60 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.1.17 #define NGM_UBT_NODE_SET_QLEN 3

Definition at line 64 of file ng_ubt.h.

Referenced by ng_ubt_rcvmsg().

7.51.2 Typedef Documentation**7.51.2.1 typedef u_int16_t ng_ubt_node_debug_ep**

Definition at line 62 of file ng_ubt.h.

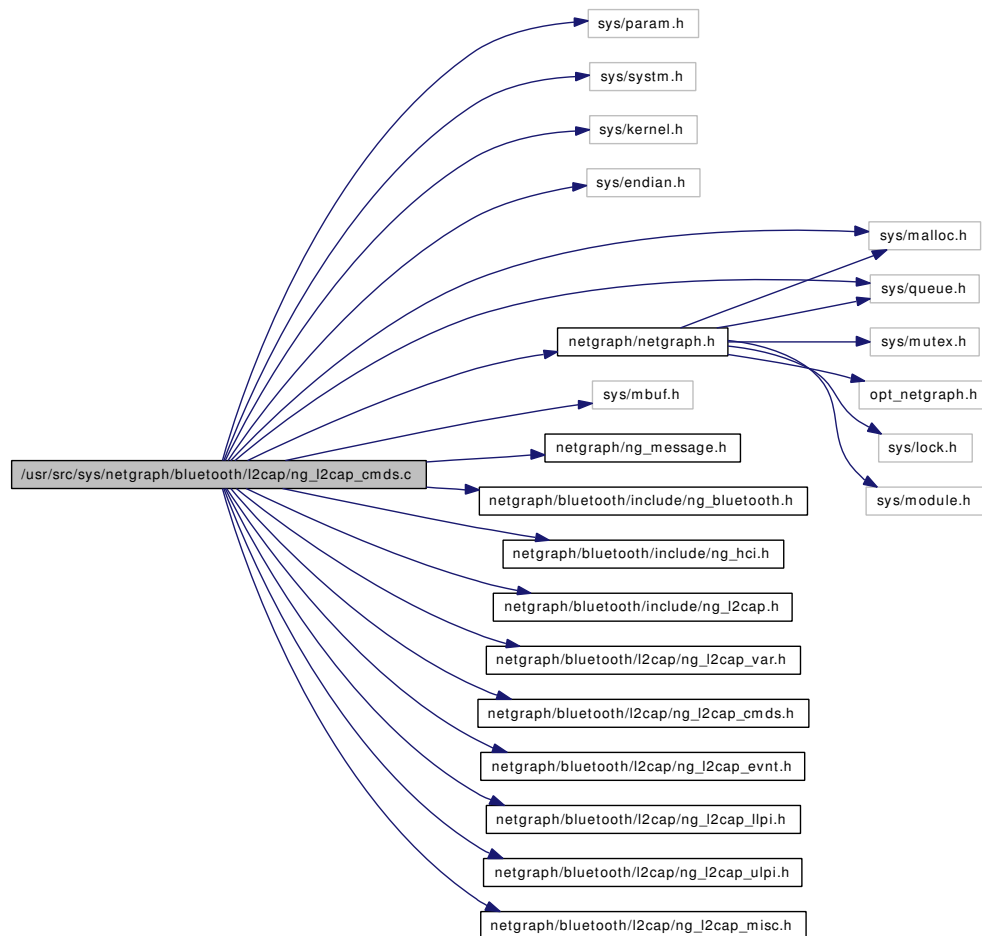
7.51.2.2 typedef u_int16_t ng_ubt_node_dev_nodes_ep

Definition at line 88 of file ng_ubt.h.

7.52 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
```

Include dependency graph for ng_l2cap_cmds.c:



Functions

- void [ng_l2cap_con_wakeup](#) ([ng_l2cap_con_p](#) con)
- void [ng_l2cap_con_fail](#) ([ng_l2cap_con_p](#) con, [u_int16_t](#) result)
- void [ng_l2cap_process_command_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)

7.52.1 Function Documentation

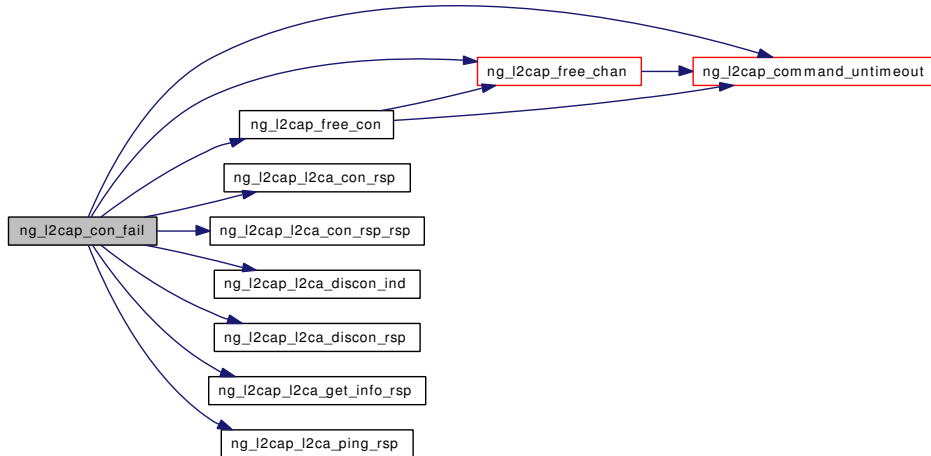
7.52.1.1 void [ng_l2cap_con_fail](#) ([ng_l2cap_con_p](#) con, [u_int16_t](#) result)

Definition at line 219 of file [ng_l2cap_cmds.c](#).

References [ng_l2cap_chan::con](#), [ng_l2cap_con::flags](#), [ng_l2cap_con::l2cap](#), [NG_L2CAP_CFG_REQ](#), [NG_L2CAP_CFG_RSP](#), [NG_L2CAP_CMD_PENDING](#), [NG_L2CAP_CMD_REJ](#), [ng_l2cap_command_timeout\(\)](#), [NG_L2CAP_CON_DYING](#), [NG_L2CAP_CON_REQ](#), [NG_L2CAP_CON_RSP](#), [NG_L2CAP_DISCON_REQ](#), [NG_L2CAP_DISCON_RSP](#), [NG_L2CAP_ECHO_REQ](#), [NG_L2CAP_ECHO_RSP](#), [ng_l2cap_free_chan\(\)](#), [ng_l2cap_free_cmd](#), [ng_l2cap_free_con\(\)](#), [NG_L2CAP_INFO](#), [NG_L2CAP_INFO_REQ](#), [NG_L2CAP_INFO_RSP](#), [ng_l2cap_l2ca_con_rsp\(\)](#), [ng_l2cap_l2ca_con_rsp_rsp\(\)](#), [ng_l2cap_l2ca_discon_ind\(\)](#), [ng_l2cap_l2ca_discon_rsp\(\)](#), [ng_l2cap_l2ca_get_info_rsp\(\)](#), [ng_l2cap_l2ca_ping_rsp\(\)](#), [NG_L2CAP_SUCCESS](#), [ng_l2cap_unlink_cmd](#), [NG_NODE_NAME](#), [NGM_L2CAP_L2CA_WRITE](#), and [ng_l2cap::node](#).

Referenced by `ng_l2cap_cleanup()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_discon_ind()`, and `ng_l2cap_process_lp_timeout()`.

Here is the call graph for this function:



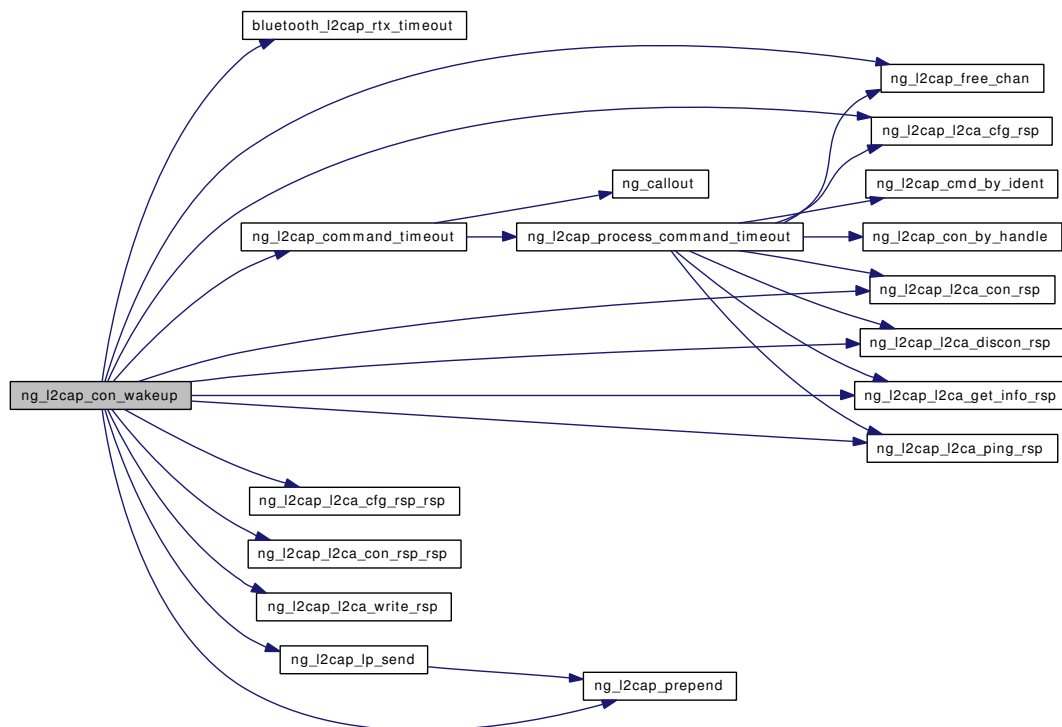
7.52.1.2 void ng_l2cap_con_wakeup (ng_l2cap_con_p con)

Definition at line 64 of file `ng_l2cap_cmds.c`.

References `bluetooth_l2cap_rtx_timeout()`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CFG_REQ`, `NG_L2CAP_CFG_RSP`, `NG_L2CAP_CLT_CID`, `NG_L2CAP_CMD_PENDING`, `NG_L2CAP_CMD_REJ`, `ng_l2cap_command_timeout()`, `NG_L2CAP_CON_REQ`, `NG_L2CAP_CON_RSP`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_DISCON_RSP`, `NG_L2CAP_ECHO_REQ`, `NG_L2CAP_ECHO_RSP`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `NG_L2CAP_INFO_REQ`, `NG_L2CAP_INFO_RSP`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lp_send()`, `NG_L2CAP_NO_RESOURCES`, `ng_l2cap_prepend()`, `NG_L2CAP_SIGNAL_CID`, `NG_L2CAP_SUCCESS`, `ng_l2cap_unlink_cmd`, `NG_NODE_NAME`, `NGM_L2CAP_L2CA_WRITE`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_lp_deliver()`.

Here is the call graph for this function:



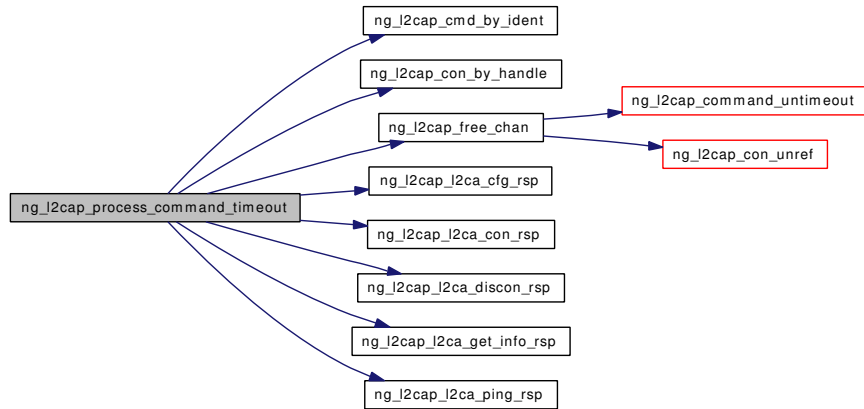
7.52.1.3 void ng_l2cap_process_command_timeout (node_p node, hook_p hook, void * arg1, int arg2)

Definition at line 316 of file ng_l2cap_cmds.c.

References NG_L2CAP_ALERT, NG_L2CAP_CFG_REQ, ng_l2cap_cmd_by_ident(), NG_L2CAP_CMD_PENDING, ng_l2cap_con_by_handle(), NG_L2CAP_CON_REQ, NG_L2CAP_DISCON_REQ, NG_L2CAP_ECHO_REQ, ng_l2cap_free_chan(), ng_l2cap_free_cmd, NG_L2CAP_INFO_REQ, ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), NG_L2CAP_TIMEOUT, ng_l2cap_unlink_cmd, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, and ng_l2cap::node.

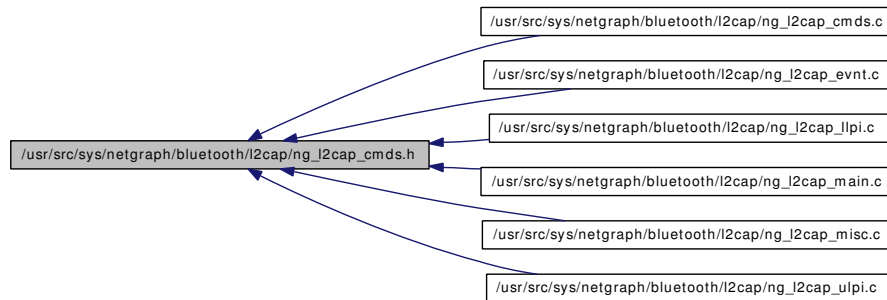
Referenced by ng_l2cap_command_timeout().

Here is the call graph for this function:



7.53 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define _ng_l2cap_cmd_rej(_m, _ident, _reason, _mtu, _scid, _dcid)`
- `#define _ng_l2cap_con_req(_m, _ident, _psm, _scid)`
- `#define _ng_l2cap_con_rsp(_m, _ident, _dcid, _scid, _result, _status)`
- `#define _ng_l2cap_cfg_req(_m, _ident, _dcid, _flags, _data)`
- `#define _ng_l2cap_cfg_rsp(_m, _ident, _scid, _flags, _result, _data)`
- `#define _ng_l2cap_build_cfg_options(_m, _mtu, _flush_timo, _flow)`
- `#define _ng_l2cap_discon_req(_m, _ident, _dcid, _scid)`
- `#define _ng_l2cap_discon_rsp(_m, _ident, _dcid, _scid)`
- `#define _ng_l2cap_echo_req(_m, _ident, _data, _size)`
- `#define _ng_l2cap_info_req(_m, _ident, _type)`
- `#define _ng_l2cap_info_rsp(_m, _ident, _type, _result, _mtu)`

Functions

- void `ng_l2cap_con_wakeup` (`ng_l2cap_con_p`)
- void `ng_l2cap_con_fail` (`ng_l2cap_con_p`, `u_int16_t`)
- void `ng_l2cap_process_command_timeout` (`node_p`, `hook_p`, `void *`, `int`)

7.53.1 Define Documentation

7.53.1.1 `#define _ng_l2cap_build_cfg_options(_m, _mtu, _flush_timo, _flow)`

Definition at line 203 of file `ng_l2cap_cmds.h`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, and `ng_l2cap_l2ca_cfg_rsp_req()`.

7.53.1.2 `#define _ng_l2cap_cfg_req(_m, _ident, _dcid, _flags, _data)`

Definition at line 134 of file `ng_l2cap_cmds.h`.

Referenced by `ng_l2cap_l2ca_cfg_req()`.

7.53.1.3 #define _ng_l2cap_cfg_rsp(_m, _ident, _scid, _flags, _result, _data)

Definition at line 168 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_cfg_rsp_req(), and send_l2cap_cfg_rsp().

7.53.1.4 #define _ng_l2cap_cmd_rej(_m, _ident, _reason, _mtu, _scid, _dcid)

Definition at line 51 of file ng_l2cap_cmds.h.

Referenced by send_l2cap_reject().

7.53.1.5 #define _ng_l2cap_con_req(_m, _ident, _psm, _scid)**Value:**

```
do {
    struct _con_req {
        ng_l2cap_cmd_hdr_t    hdr;
        ng_l2cap_con_req_cp   param;
    } __attribute__((packed)) *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), struct _con_req *);
    c->hdr.code = NG_L2CAP_CON_REQ;
    c->hdr.ident = (_ident);
    c->hdr.length = htobe16(sizeof(c->param));

    c->param.psm = htobe16((_psm));
    c->param.scid = htobe16((_scid));
} while (0)
```

Definition at line 86 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_con_req().

7.53.1.6 #define _ng_l2cap_con_rsp(_m, _ident, _dcid, _scid, _result, _status)**Value:**

```
do {
    struct _con_rsp {
        ng_l2cap_cmd_hdr_t    hdr;
        ng_l2cap_con_rsp_cp   param;
    } __attribute__((packed)) *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), struct _con_rsp *);
    c->hdr.code = NG_L2CAP_CON_RSP;
    c->hdr.ident = (_ident);
```

```

    c->hdr.length = htobe16(sizeof(c->param));
    c->param.dcid = htobe16((_dcid));
    c->param.scid = htobe16((_scid));
    c->param.result = htobe16((_result));
    c->param.status = htobe16((_status));
} while (0)

```

Definition at line 109 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_con_rsp_req(), and send_l2cap_con_rej().

7.53.1.7 #define ng_l2cap_discon_req(_m, _ident, _dcid, _scid)

Value:

```

do {
    struct _discon_req {
        ng_l2cap_cmd_hdr_t    hdr;
        ng_l2cap_discon_req_cp param;
    } __attribute__((packed)) *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), struct _discon_req *);
    c->hdr.code = NG_L2CAP_DISCON_REQ;
    c->hdr.ident = (_ident);
    c->hdr.length = htobe16(sizeof(c->param));

    c->param.dcid = htobe16((_dcid));
    c->param.scid = htobe16((_scid));
} while (0)

```

Definition at line 277 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_discon_req().

7.53.1.8 #define ng_l2cap_discon_rsp(_m, _ident, _dcid, _scid)

Value:

```

do {
    struct _discon_rsp {
        ng_l2cap_cmd_hdr_t    hdr;
        ng_l2cap_discon_rsp_cp param;
    } __attribute__((packed)) *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), struct _discon_rsp *);
    c->hdr.code = NG_L2CAP_DISCON_RSP;
    c->hdr.ident = (_ident);
    c->hdr.length = htobe16(sizeof(c->param));
} while (0)

```

```

        c->param.dcid = htobe16((_dcid));
        c->param.scid = htobe16((_scid));
    } while (0)

```

Definition at line 300 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_process_discon_req().

7.53.1.9 #define ng_l2cap_echo_req(_m, _ident, _data, _size)

Value:

```

do {
    ng_l2cap_cmd_hdr_t    *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), ng_l2cap_cmd_hdr_t *);
    c->code = NG_L2CAP_ECHO_REQ;
    c->ident = (_ident);
    c->length = 0;

    if ((_data) != NULL) {
        m_copyback((_m), sizeof(*c), (_size), (_data));
        c->length += (_size);
    }

    c->length = htobe16(c->length);
} while (0)

```

Definition at line 323 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_ping_req().

7.53.1.10 #define ng_l2cap_info_req(_m, _ident, _type)

Value:

```

do {
    struct _info_req {
        ng_l2cap_cmd_hdr_t    hdr;
        ng_l2cap_info_req_cp  param;
    } __attribute__((packed)) *c = NULL;

    MGETHDR((_m), M_DONTWAIT, MT_DATA);
    if ((_m) == NULL)
        break;

    (_m)->m_pkthdr.len = (_m)->m_len = sizeof(*c);

    c = mtod((_m), struct _info_req *);
    c->hdr.code = NG_L2CAP_INFO_REQ;
    c->hdr.ident = (_ident);
    c->hdr.length = htobe16(sizeof(c->param));

    c->param.type = htobe16((_type));
} while (0)

```

Definition at line 347 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_l2ca_get_info_req().

7.53.1.11 #define ng_l2cap_info_rsp(*m*, *ident*, *type*, *result*, *mtu*)

Definition at line 369 of file ng_l2cap_cmds.h.

Referenced by ng_l2cap_process_info_req().

7.53.2 Function Documentation

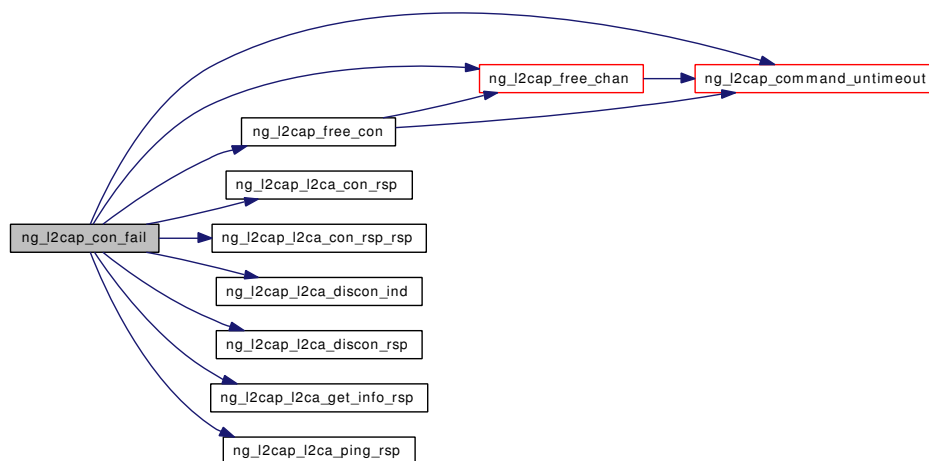
7.53.2.1 void ng_l2cap_con_fail([ng_l2cap_con_p](#), [u_int16_t](#))

Definition at line 219 of file ng_l2cap_cmds.c.

References [ng_l2cap_chan::con](#), [ng_l2cap_con::flags](#), [ng_l2cap_con::l2cap](#), [NG_L2CAP_CFG_REQ](#), [NG_L2CAP_CFG_RSP](#), [NG_L2CAP_CMD_PENDING](#), [NG_L2CAP_CMD_REJ](#), [ng_l2cap_command_timeout\(\)](#), [NG_L2CAP_CON_DYING](#), [NG_L2CAP_CON_REQ](#), [NG_L2CAP_CON_RSP](#), [NG_L2CAP_DISCON_REQ](#), [NG_L2CAP_DISCON_RSP](#), [NG_L2CAP_ECHO_REQ](#), [NG_L2CAP_ECHO_RSP](#), [ng_l2cap_free_chan\(\)](#), [ng_l2cap_free_cmd](#), [ng_l2cap_free_con\(\)](#), [NG_L2CAP_INFO](#), [NG_L2CAP_INFO_REQ](#), [NG_L2CAP_INFO_RSP](#), [ng_l2cap_l2ca_con_rsp\(\)](#), [ng_l2cap_l2ca_con_rsp_rsp\(\)](#), [ng_l2cap_l2ca_discon_ind\(\)](#), [ng_l2cap_l2ca_discon_rsp\(\)](#), [ng_l2cap_l2ca_get_info_rsp\(\)](#), [ng_l2cap_l2ca_ping_rsp\(\)](#), [NG_L2CAP_SUCCESS](#), [ng_l2cap_unlink_cmd](#), [NG_NODE_NAME](#), [NGM_L2CAP_L2CA_WRITE](#), and [ng_l2cap::node](#).

Referenced by [ng_l2cap_cleanup\(\)](#), [ng_l2cap_lp_con_cfm\(\)](#), [ng_l2cap_lp_discon_ind\(\)](#), and [ng_l2cap_process_lp_timeout\(\)](#).

Here is the call graph for this function:



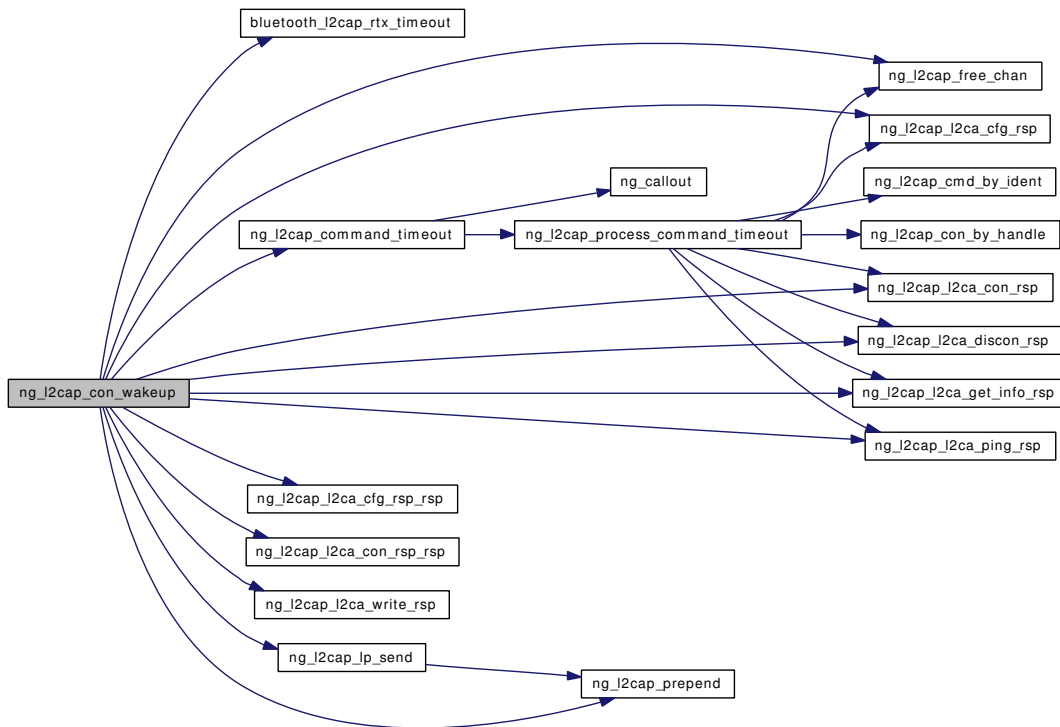
7.53.2.2 void ng_l2cap_con_wakeup([ng_l2cap_con_p](#))

Definition at line 64 of file ng_l2cap_cmds.c.

References `bluetooth_l2cap_rtx_timeout()`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CFG_REQ`, `NG_L2CAP_CFG_RSP`, `NG_L2CAP_CLT_CID`, `NG_L2CAP_CMD_PENDING`, `NG_L2CAP_CMD_REJ`, `ng_l2cap_command_timeout()`, `NG_L2CAP_CON_REQ`, `NG_L2CAP_CON_RSP`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_DISCON_RSP`, `NG_L2CAP_ECHO_REQ`, `NG_L2CAP_ECHO_RSP`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `NG_L2CAP_INFO_REQ`, `NG_L2CAP_INFO_RSP`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lp_send()`, `NG_L2CAP_NO_RESOURCES`, `ng_l2cap_prepend()`, `NG_L2CAP_SIGNAL_CID`, `NG_L2CAP_SUCCESS`, `ng_l2cap_unlink_cmd`, `NG_NODE_NAME`, `NGM_L2CAP_L2CA_WRITE`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_lp_deliver()`.

Here is the call graph for this function:



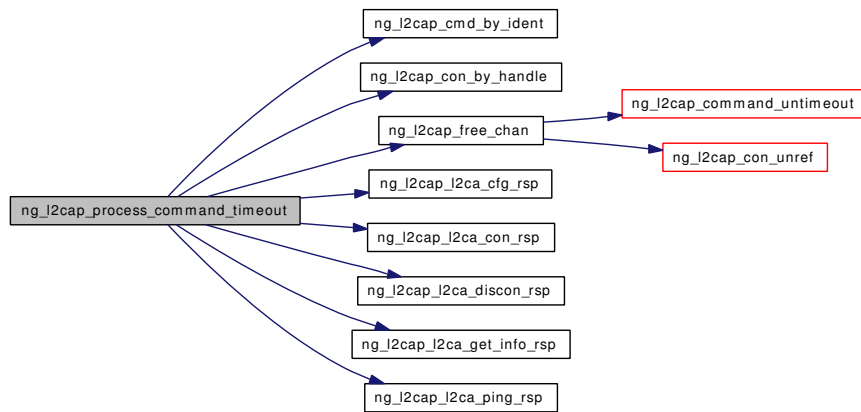
7.53.2.3 void ng_l2cap_process_command_timeout (node_p, hook_p, void *, int)

Definition at line 316 of file `ng_l2cap_cmds.c`.

References `NG_L2CAP_ALERT`, `NG_L2CAP_CFG_REQ`, `ng_l2cap_cmd_by_ident()`, `NG_L2CAP_CMD_PENDING`, `ng_l2cap_con_by_handle()`, `NG_L2CAP_CON_REQ`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_ECHO_REQ`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `NG_L2CAP_INFO_REQ`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `NG_L2CAP_TIMEOUT`, `ng_l2cap_unlink_cmd`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_command_timeout()`.

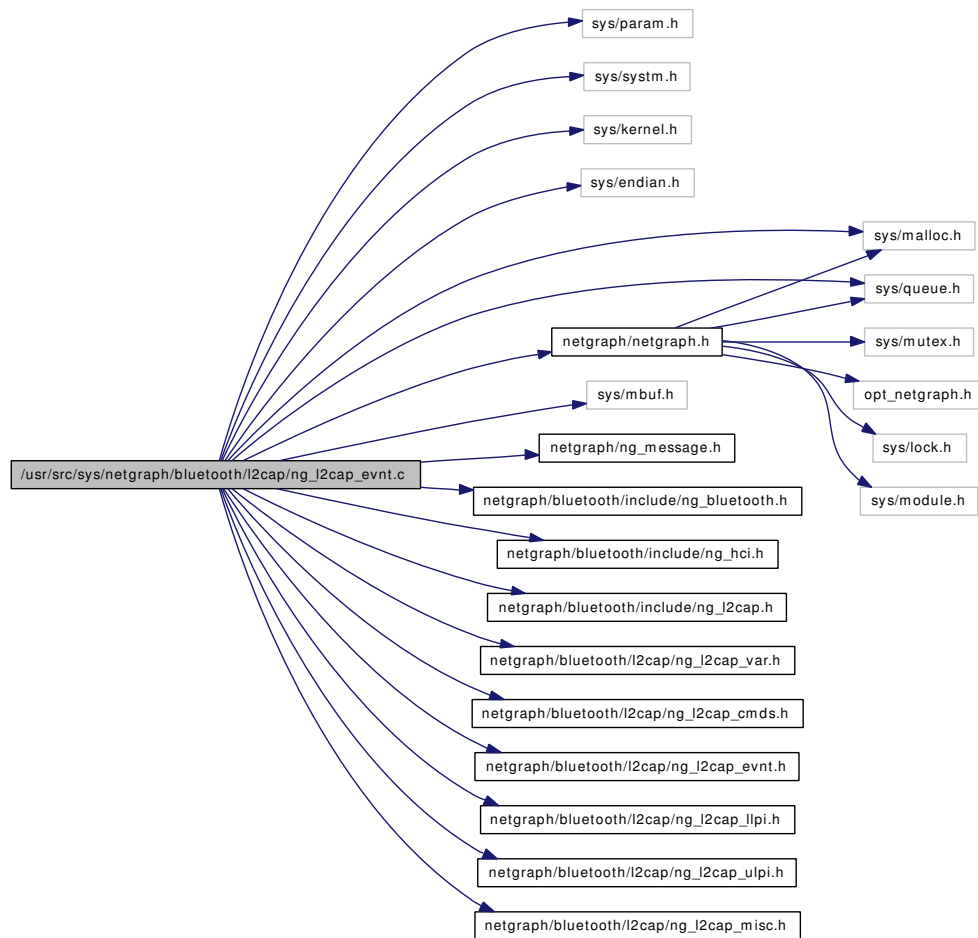
Here is the call graph for this function:



7.54 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
```

Include dependency graph for ng_l2cap_evnt.c:



Functions

- static int [ng_l2cap_process_signal_cmd](#) (ng_l2cap_con_p)
- static int [ng_l2cap_process_cmd_rej](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_con_req](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_con_rsp](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_cfg_req](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_cfg_rsp](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_discon_req](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_discon_rsp](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_echo_req](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_echo_rsp](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_info_req](#) (ng_l2cap_con_p, u_int8_t)
- static int [ng_l2cap_process_info_rsp](#) (ng_l2cap_con_p, u_int8_t)
- static int [send_l2cap_reject](#) (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, u_int16_t)
- static int [send_l2cap_con_rej](#) (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, u_int16_t)
- static int [send_l2cap_cfg_rsp](#) (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, struct mbuf *)
- static int [get_next_l2cap_opt](#) (struct mbuf *, int *, ng_l2cap_cfg_opt_p, ng_l2cap_cfg_opt_val_p)
- int [ng_l2cap_receive](#) (ng_l2cap_con_p con)

7.54.1 Function Documentation

7.54.1.1 `static int get_next_l2cap_opt (struct mbuf *, int *, ng_l2cap_cfg_opt_p, ng_l2cap_cfg_opt_val_p)` [static]

Definition at line 1266 of file `ng_l2cap_evnt.c`.

References `NG_L2CAP_OPT_FLUSH_TIMO`, `NG_L2CAP_OPT_FLUSH_TIMO_SIZE`, `NG_L2CAP_OPT_HINT`, `NG_L2CAP_OPT_HINT_MASK`, `NG_L2CAP_OPT_MTU`, `NG_L2CAP_OPT_MTU_SIZE`, `NG_L2CAP_OPT_QOS`, and `NG_L2CAP_OPT_QOS_SIZE`.

Referenced by `ng_l2cap_process_cfg_req()`, and `ng_l2cap_process_cfg_rsp()`.

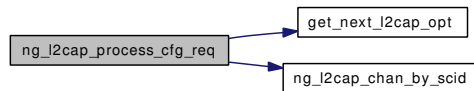
7.54.1.2 `static int ng_l2cap_process_cfg_req (ng_l2cap_con_p, u_int8_t)` [static]

Definition at line 521 of file `ng_l2cap_evnt.c`.

References `ng_l2cap_chan::cfg_state`, `ng_l2cap_cfg_opt_val_t::flow`, `ng_l2cap_cfg_opt_val_t::flush_timo`, `ng_l2cap_chan::flush_timo`, `get_next_l2cap_opt()`, `ng_l2cap_chan::iflow`, `ng_l2cap_con::l2cap`, `ng_l2cap_cfg_opt_val_t::mtu`, `NG_FREE_M`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_ERR`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_OPEN`, `NG_L2CAP_OPT_CFLAG`, `NG_L2CAP_OPT_FLUSH_TIMO`, `NG_L2CAP_OPT_MTU`, `NG_L2CAP_OPT_QOS`, `NG_L2CAP_REJECT`, `NG_L2CAP_UNKNOWN_OPTION`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::omtu`, `ng_l2cap_con::rx_pkt`, and `ng_l2cap_chan::state`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



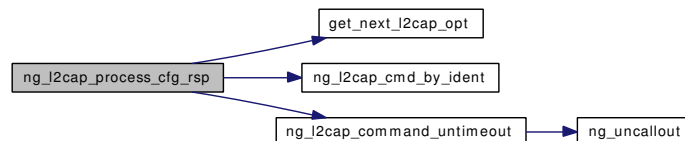
7.54.1.3 `static int ng_l2cap_process_cfg_rsp (ng_l2cap_con_p, u_int8_t)` [static]

Definition at line 655 of file `ng_l2cap_evnt.c`.

References `ng_l2cap_con::con_handle`, `ng_l2cap_cfg_opt_val_t::flow`, `ng_l2cap_cfg_opt_val_t::flush_timo`, `get_next_l2cap_opt()`, `ng_l2cap_con::l2cap`, `ng_l2cap_cfg_opt_val_t::mtu`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `ng_l2cap_cmd_by_ident()`, `ng_l2cap_command_untimeout()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_ERR`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_OPT_CFLAG`, `NG_L2CAP_OPT_FLUSH_TIMO`, `NG_L2CAP_OPT_MTU`, `NG_L2CAP_OPT_QOS`, `NG_L2CAP_UNKNOWN`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



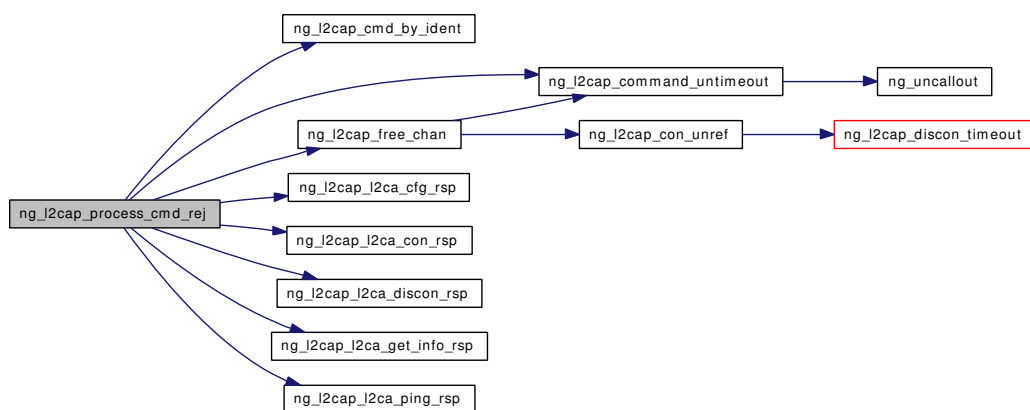
7.54.1.4 static int ng_l2cap_process_cmd_rej (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 273 of file ng_l2cap_evnt.c.

References ng_l2cap_con::l2cap, NG_FREE_M, NG_L2CAP_ALERT, NG_L2CAP_CFG_REQ, ng_l2cap_cmd_by_ident(), ng_l2cap_command_untimeout(), NG_L2CAP_CON_REQ, NG_L2CAP_DISCON_REQ, NG_L2CAP_ECHO_REQ, NG_L2CAP_ERR, ng_l2cap_free_chan(), ng_l2cap_free_cmd, NG_L2CAP_INFO_REQ, ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), NG_L2CAP_M_PULLUP, ng_l2cap_unlink_cmd, NG_NODE_NAME, ng_l2cap::node, and ng_l2cap_con::rx_pkt.

Referenced by ng_l2cap_process_signal_cmd().

Here is the call graph for this function:



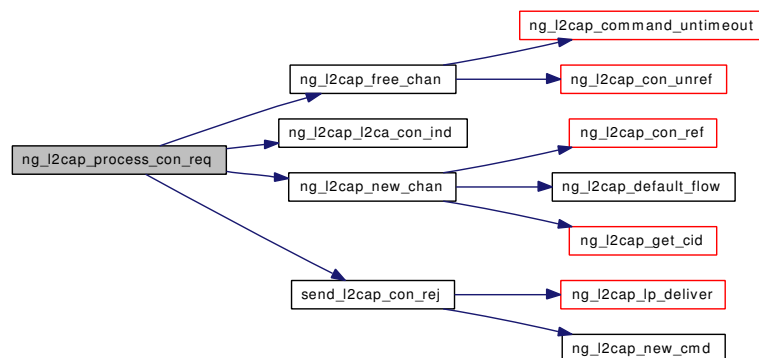
7.54.1.5 static int ng_l2cap_process_con_req (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 347 of file ng_l2cap_evnt.c.

References ng_l2cap_chan::dcid, ng_l2cap_chan::ident, ng_l2cap_con::l2cap, NG_FREE_M, ng_l2cap_free_chan(), ng_l2cap_l2ca_con_ind(), NG_L2CAP_M_PULLUP, ng_l2cap_new_chan(), NG_L2CAP_NO_RESOURCES, NG_L2CAP_PSM_NOT_SUPPORTED, NG_L2CAP_W4_L2CA_CON_RSP, ng_l2cap_con::rx_pkt, ng_l2cap_chan::scid, send_l2cap_con_rej(), and ng_l2cap_chan::state.

Referenced by ng_l2cap_process_signal_cmd().

Here is the call graph for this function:



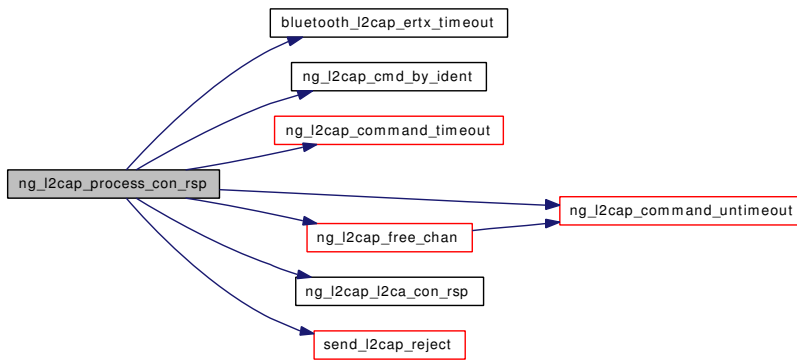
7.54.1.6 static int ng_l2cap_process_con_rsp (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 401 of file ng_l2cap_evnt.c.

References bluetooth_l2cap_ertx_timeout(), ng_l2cap_con::con_handle, ng_l2cap_con::l2cap, NG_FREE_M, ng_l2cap_cmd_by_ident(), ng_l2cap_command_timeout(), ng_l2cap_command_untimeout(), NG_L2CAP_CONFIG, NG_L2CAP_ERR, ng_l2cap_free_chan(), ng_l2cap_free_cmd, NG_L2CAP_INFO, ng_l2cap_l2ca_con_rsp(), NG_L2CAP_M_PULLUP, NG_L2CAP_PENDING, NG_L2CAP_REJ_INVALID_CID, NG_L2CAP_SUCCESS, ng_l2cap_unlink_cmd, NG_L2CAP_W4_L2CAP_CON_RSP, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, and send_l2cap_reject().

Referenced by ng_l2cap_process_signal_cmd().

Here is the call graph for this function:



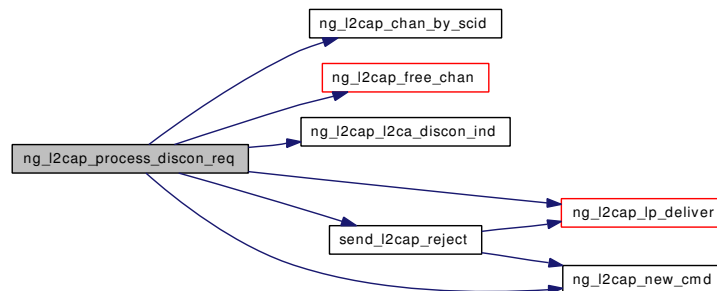
7.54.1.7 static int ng_l2cap_process_discon_req (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 809 of file ng_l2cap_evnt.c.

References _ng_l2cap_discon_rsp, ng_l2cap_chan::dcid, ng_l2cap_con::l2cap, NG_FREE_M, ng_l2cap_chan_by_scid(), NG_L2CAP_CONFIG, NG_L2CAP_DISCON_RSP, NG_L2CAP_ERR, ng_l2cap_free_chan(), ng_l2cap_free_cmd, ng_l2cap_l2ca_discon_ind(), ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), NG_L2CAP_M_PULLUP, ng_l2cap_new_cmd(), NG_L2CAP_OPEN, NG_L2CAP_REJ_INVALID_CID, NG_L2CAP_W4_L2CAP_DISCON_RSP, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, ng_l2cap_chan::scid, send_l2cap_reject(), and ng_l2cap_chan::state.

Referenced by ng_l2cap_process_signal_cmd().

Here is the call graph for this function:



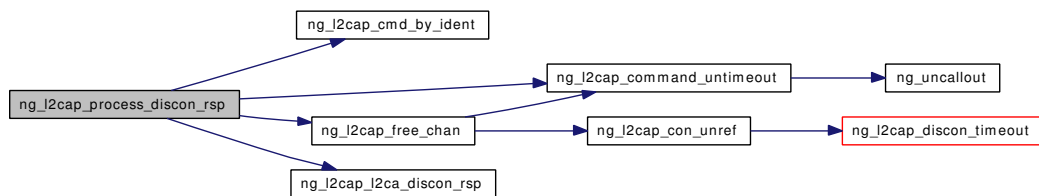
7.54.1.8 `static int ng_l2cap_process_discon_rsp (ng_l2cap_con_p, u_int8_t)` [static]

Definition at line 900 of file ng_l2cap_evnt.c.

References `ng_l2cap_con::con_handle`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `ng_l2cap_cmd_by_ident()`, `ng_l2cap_command_untimeout()`, `NG_L2CAP_ERR`, `ng_l2cap_free_chan()`, `ng_l2cap_l2ca_discon_rsp()`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_SUCCESS`, `NG_L2CAP_W4_L2CAP_DISCON_RSP`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:

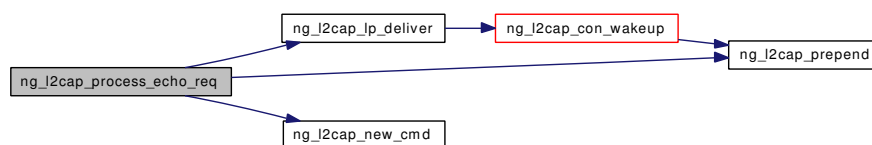
**7.54.1.9** `static int ng_l2cap_process_echo_req (ng_l2cap_con_p, u_int8_t)` [static]

Definition at line 969 of file ng_l2cap_evnt.c.

References `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `NG_L2CAP_ECHO_RSP`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `ng_l2cap_prepend()`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:

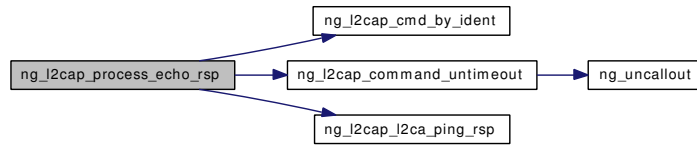
**7.54.1.10** `static int ng_l2cap_process_echo_rsp (ng_l2cap_con_p, u_int8_t)` [static]

Definition at line 1010 of file ng_l2cap_evnt.c.

References `ng_l2cap_con::l2cap`, `NG_FREE_M`, `ng_l2cap_cmd_by_ident()`, `ng_l2cap_command_untimeout()`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_l2ca_ping_rsp()`, `NG_L2CAP_SUCCESS`, `ng_l2cap_unlink_cmd`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



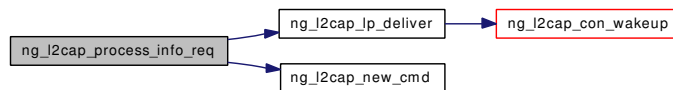
7.54.1.11 static int ng_l2cap_process_info_req (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 1048 of file ng_l2cap_evnt.c.

References `_ng_l2cap_info_rsp`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_L2CAP_CONNLESS_MTU`, `ng_l2cap_free_cmd`, `NG_L2CAP_INFO_RSP`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_MTU_DEFAULT`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NOT_SUPPORTED`, `NG_L2CAP_SUCCESS`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



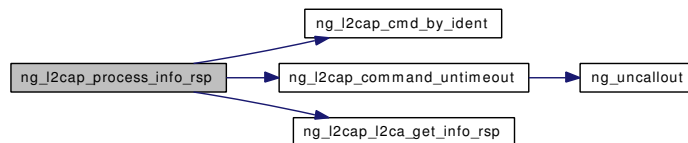
7.54.1.12 static int ng_l2cap_process_info_rsp (ng_l2cap_con_p, u_int8_t) [static]

Definition at line 1096 of file ng_l2cap_evnt.c.

References `ng_l2cap_con::l2cap`, `NG_FREE_M`, `ng_l2cap_cmd_by_ident()`, `ng_l2cap_command_untimeout()`, `NG_L2CAP_CONNLESS_MTU`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_l2ca_get_info_rsp()`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_SUCCESS`, `NG_L2CAP_UNKNOWN`, `ng_l2cap_unlink_cmd`, `NG_L2CAP_WARN`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



7.54.1.13 static int ng_l2cap_process_signal_cmd (ng_l2cap_con_p) [static]

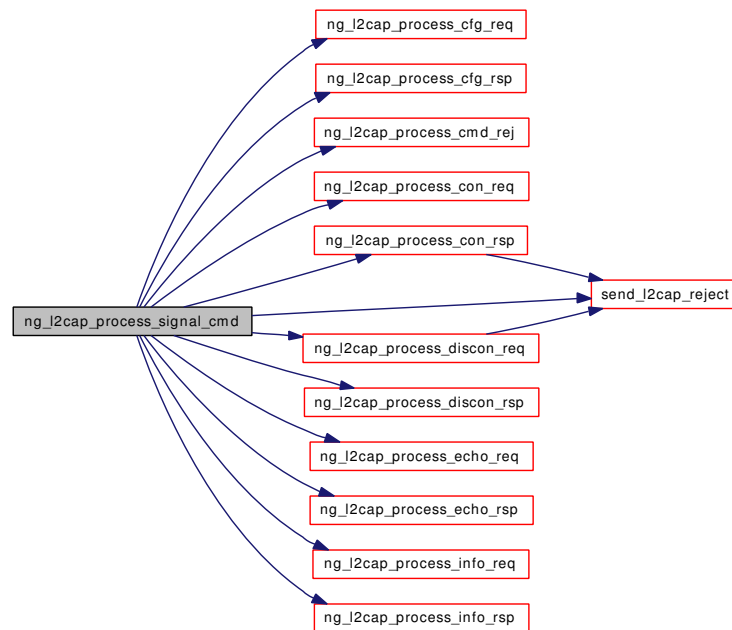
Definition at line 153 of file ng_l2cap_evnt.c.

References `ng_l2cap_con::ident`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_L2CAP_CFG_REQ`, `NG_L2CAP_CFG_RSP`, `NG_L2CAP_CMD_REJ`, `NG_L2CAP_CON_REQ`, `NG_L2CAP_CON_RSP`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_DISCON_RSP`, `NG_L2CAP_ECHO_REQ`, `NG_L2CAP_ECHO_RSP`, `NG_L2CAP_ERR`, `NG_L2CAP_INFO_REQ`, `NG_L2CAP_INFO_RSP`, `NG_L2CAP_M_PULLUP`,

ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_echo_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), NG_L2CAP_REJ_NOT_UNDERSTOOD, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, and send_l2cap_reject().

Referenced by ng_l2cap_receive().

Here is the call graph for this function:



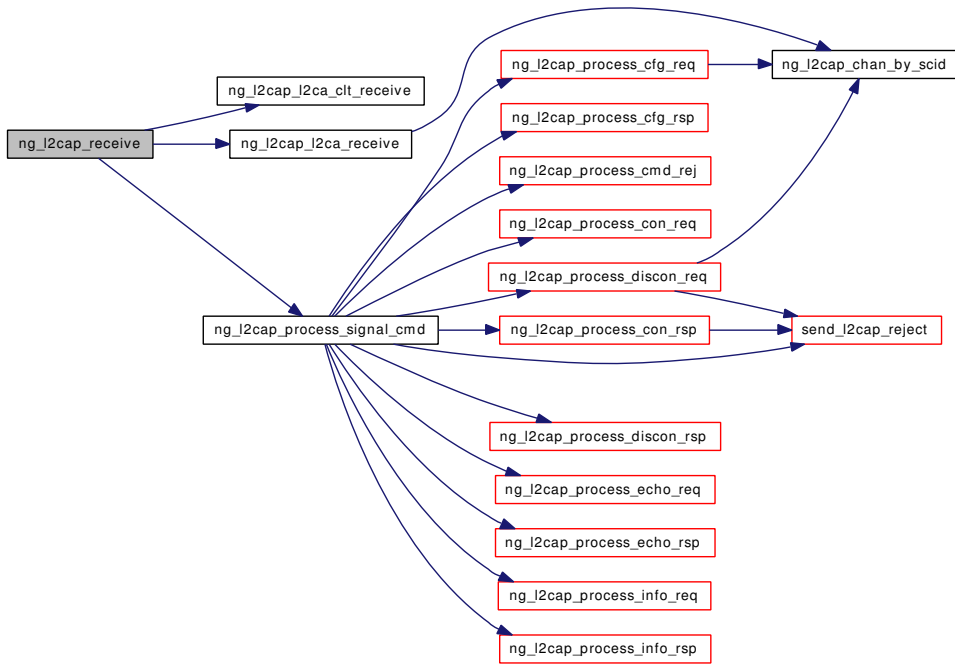
7.54.1.14 int ng_l2cap_receive (ng_l2cap_con_p con)

Definition at line 86 of file ng_l2cap_evnt.c.

References ng_l2cap_con::l2cap, NG_FREE_M, NG_L2CAP_CLT_CID, NG_L2CAP_ERR, ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_receive(), NG_L2CAP_M_PULLUP, ng_l2cap_process_signal_cmd(), NG_L2CAP_SIGNAL_CID, NG_NODE_NAME, ng_l2cap::node, and ng_l2cap_con::rx_pkt.

Referenced by ng_l2cap_lp_receive().

Here is the call graph for this function:

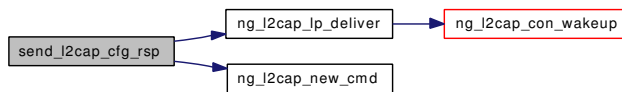


7.54.1.15 static int send_l2cap_cfg_rsp (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, struct mbuf *) [static]

Definition at line 1228 of file ng_l2cap_evnt.c.

References _ng_l2cap_cfg_rsp, NG_FREE_M, NG_L2CAP_CFG_RSP, ng_l2cap_free_cmd, ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), and ng_l2cap_new_cmd().

Here is the call graph for this function:



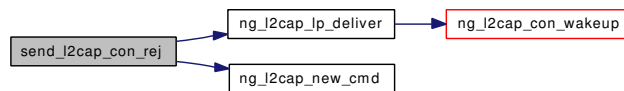
7.54.1.16 static int send_l2cap_con_rej (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, u_int16_t) [static]

Definition at line 1200 of file ng_l2cap_evnt.c.

References _ng_l2cap_con_rsp, NG_L2CAP_CON_RSP, ng_l2cap_free_cmd, ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), and ng_l2cap_new_cmd().

Referenced by ng_l2cap_process_con_req().

Here is the call graph for this function:



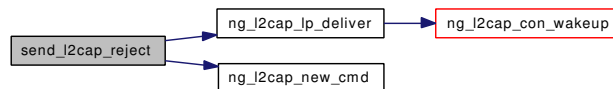
7.54.1.17 `static int send_l2cap_reject (ng_l2cap_con_p, u_int8_t, u_int16_t, u_int16_t, u_int16_t, u_int16_t)` [static]

Definition at line 1172 of file `ng_l2cap_evnt.c`.

References `_ng_l2cap_cmd_rej`, `NG_L2CAP_CMD_REJ`, `ng_l2cap_free_cmd`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, and `ng_l2cap_new_cmd()`.

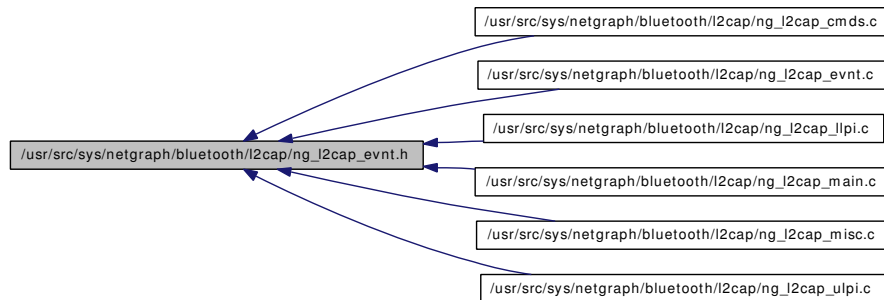
Referenced by `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_req()`, and `ng_l2cap_process_signal_cmd()`.

Here is the call graph for this function:



7.55 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evnt.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- [int ng_l2cap_receive \(ng_l2cap_con_p\)](#)

7.55.1 Function Documentation

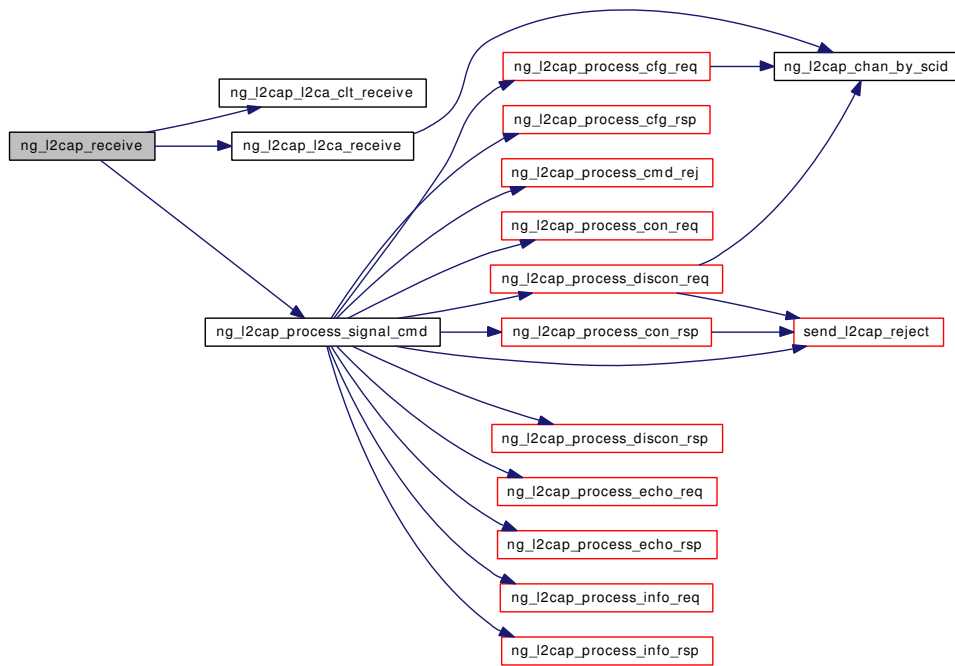
7.55.1.1 int ng_l2cap_receive (ng_l2cap_con_p)

Definition at line 86 of file ng_l2cap_evnt.c.

References [ng_l2cap_con::l2cap](#), [NG_FREE_M](#), [NG_L2CAP_CLT_CID](#), [NG_L2CAP_ERR](#), [ng_l2cap_l2ca_clt_receive\(\)](#), [ng_l2cap_l2ca_receive\(\)](#), [NG_L2CAP_M_PULLUP](#), [ng_l2cap_process_signal_cmd\(\)](#), [NG_L2CAP_SIGNAL_CID](#), [NG_NODE_NAME](#), [ng_l2cap::node](#), and [ng_l2cap_con::rx_pkt](#).

Referenced by [ng_l2cap_lp_receive\(\)](#).

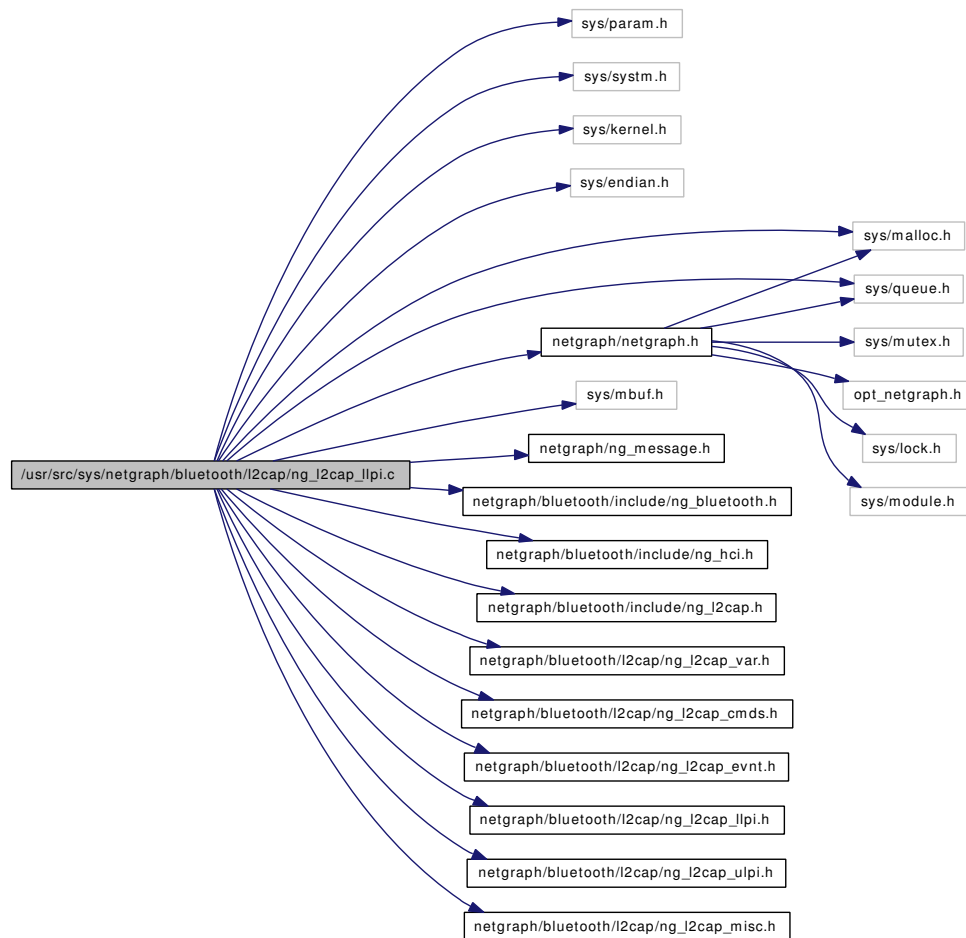
Here is the call graph for this function:



7.56 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
```

Include dependency graph for ng_l2cap_llpi.c:



Functions

- `int ng_l2cap_lp_con_req (ng_l2cap_p l2cap, bdaddr_p bdaddr)`
- `int ng_l2cap_lp_con_cfm (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_lp_con_ind (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_lp_discon_ind (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_lp_qos_req (ng_l2cap_p l2cap, u_int16_t con_handle, ng_l2cap_flow_p flow)`
- `int ng_l2cap_lp_qos_cfm (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_lp_qos_ind (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_lp_send (ng_l2cap_con_p con, u_int16_t dcid, struct mbuf *m0)`
- `int ng_l2cap_lp_receive (ng_l2cap_p l2cap, struct mbuf *m)`
- `void ng_l2cap_lp_deliver (ng_l2cap_con_p con)`
- `void ng_l2cap_process_lp_timeout (node_p node, hook_p hook, void *arg1, int con_handle)`
- `void ng_l2cap_process_discon_timeout (node_p node, hook_p hook, void *arg1, int con_handle)`

7.56.1 Function Documentation

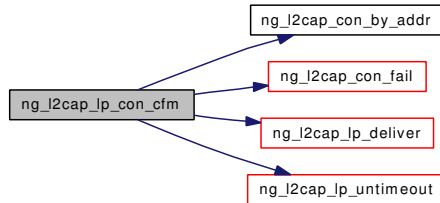
7.56.1.1 `int ng_l2cap_lp_con_cfm (ng_l2cap_p l2cap, struct ng_mesg *msg)`

Definition at line 135 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `ng_l2cap_lp_deliver()`, `ng_l2cap_lp_untimeout()`, `NG_L2CAP_W4_LP_CON_CFM`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



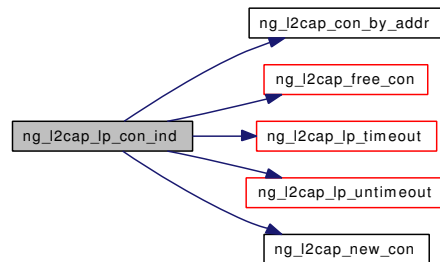
7.56.1.2 `int ng_l2cap_lp_con_ind (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 203 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap::hci`, `ng_mesg::header`, `NG_HCI_LINK_ACL`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_ERR`, `ng_l2cap_free_con()`, `NG_L2CAP_HOOK_HCI`, `ng_l2cap_lp_timeout()`, `ng_l2cap_lp_untimeout()`, `ng_l2cap_new_con()`, `NG_L2CAP_W4_LP_CON_CFM`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_CON_RSP`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



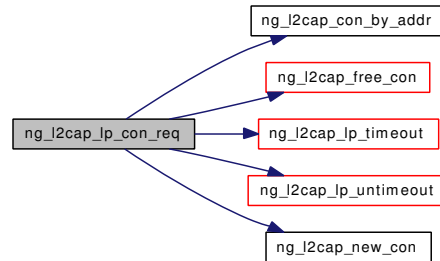
7.56.1.3 `int ng_l2cap_lp_con_req (ng_l2cap_p l2cap, bdaddr_p bdaddr)`

Definition at line 67 of file `ng_l2cap_llpi.c`.

References `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap_con::flags`, `ng_l2cap::hci`, `NG_HCI_LINK_ACL`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_CON_OUTGOING`, `NG_L2CAP_ERR`, `ng_l2cap_free_con()`, `NG_L2CAP_HOOK_HCI`, `ng_l2cap_lp_timeout()`, `ng_l2cap_lp_untimeout()`, `ng_l2cap_new_con()`, `NG_L2CAP_W4_LP_CON_CFM`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_CON_REQ`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_get_info_req()`, and `ng_l2cap_l2ca_ping_req()`.

Here is the call graph for this function:



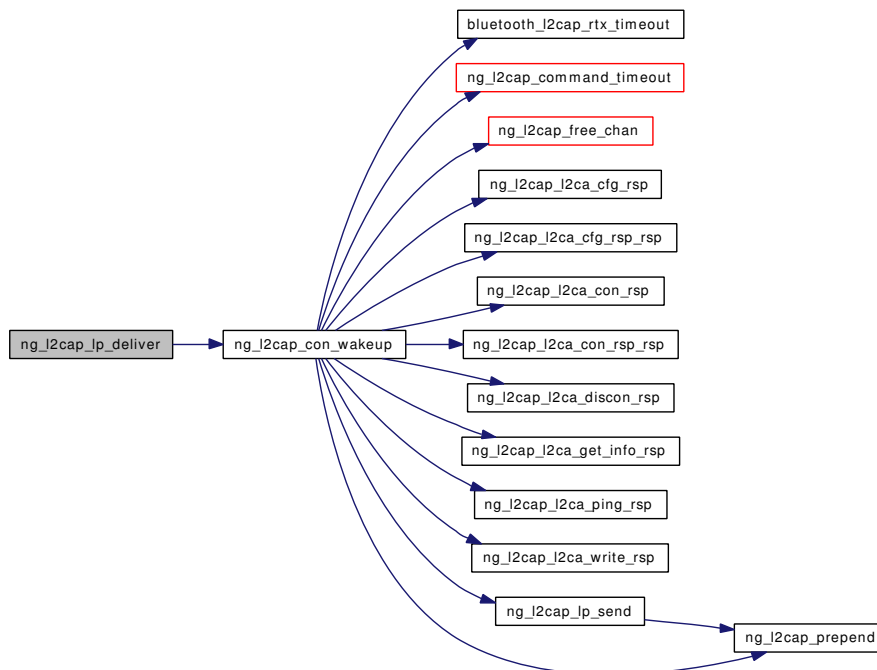
7.56.1.4 void ng_l2cap_lp_deliver (ng_l2cap_con_p con)

Definition at line 739 of file `ng_l2cap_llpi.c`.

References `ng_l2cap::hci`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CON_OPEN`, `ng_l2cap_con_wakeup()`, `NG_L2CAP_ERR`, `NG_L2CAP_HOOK_HCI`, `NG_L2CAP_INFO`, `NG_NODE_NAME`, `NG_SEND_DATA_ONLY`, `ng_l2cap::node`, `ng_l2cap_con::state`, and `ng_l2cap_con::tx_pkt`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_process_discon_req()`, `ng_l2cap_process_echo_req()`, `ng_l2cap_process_info_req()`, `send_l2cap_cfg_rsp()`, `send_l2cap_con_rej()`, and `send_l2cap_reject()`.

Here is the call graph for this function:



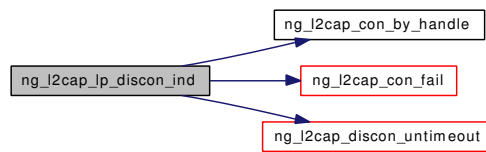
7.56.1.5 `int ng_l2cap_lp_discon_ind (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 284 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap_con::flags`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `ng_l2cap_con_by_handle()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_OPEN`, `ng_l2cap_discon_untimeout()`, `NG_L2CAP_ERR`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



7.56.1.6 `int ng_l2cap_lp_qos_cfm (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 408 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `NG_NODE_NAME`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

7.56.1.7 `int ng_l2cap_lp_qos_ind (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 435 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_handle()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



7.56.1.8 `int ng_l2cap_lp_qos_req (ng_l2cap_p l2cap, u_int16_t con_handle, ng_l2cap_flow_p flow)`

Definition at line 344 of file `ng_l2cap_llpi.c`.

References `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap::hci`, `NG_HOOK_NOT_VALID`, `ng_l2cap_con_by_handle()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `NG_L2CAP_HOOK_HCI`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_QOS_REQ`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Here is the call graph for this function:



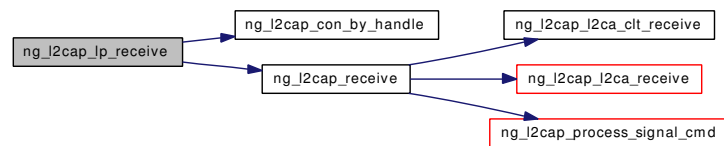
7.56.1.9 int ng_l2cap_ip_receive (ng_l2cap_p l2cap, struct mbuf * m)

Definition at line 601 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, NG_FREE_M, NG_HCI_CON_HANDLE, NG_HCI_PACKET_FRAGMENT, NG_HCI_PACKET_START, NG_HCI_PB_FLAG, NG_L2CAP_ALERT, ng_l2cap_con_by_handle(), NG_L2CAP_CON_OPEN, NG_L2CAP_ERR, NG_L2CAP_INFO, NG_L2CAP_M_PULLUP, ng_l2cap_receive(), NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, ng_l2cap_con::rx_pkt_len, and ng_l2cap_con::state.

Referenced by ng_l2cap_rcvdata().

Here is the call graph for this function:



7.56.1.10 int ng_l2cap_ip_send (ng_l2cap_con_p con, u_int16_t dcid, struct mbuf * m0)

Definition at line 485 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, ng_l2cap_con::l2cap, NG_FREE_M, NG_HCI_ACL_DATA_PKT, NG_HCI_MK_CON_HANDLE, NG_HCI_PACKET_FRAGMENT, NG_HCI_PACKET_START, NG_L2CAP_ALERT, NG_L2CAP_INFO, ng_l2cap_prepend(), NG_NODE_NAME, ng_l2cap::node, ng_l2cap::pkt_size, and ng_l2cap_con::tx_pkt.

Referenced by ng_l2cap_con_wakeup().

Here is the call graph for this function:



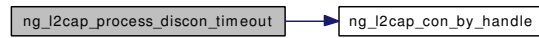
7.56.1.11 void ng_l2cap_process_discon_timeout (node_p node, hook_p hook, void * arg1, int con_handle)

Definition at line 853 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, ng_mesg::data, ng_l2cap_con::flags, ng_l2cap::hci, NG_HOOK_NOT_VALID, NG_L2CAP_ALERT, NG_L2CAP_CON_AUTO_DISCON_TIMO, ng_l2cap_con_by_handle(), NG_L2CAP_ERR, NG_L2CAP_HOOK_HCI, NG_MKMESSAGE, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_DISCON_REQ, ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_discon_timeout().

Here is the call graph for this function:



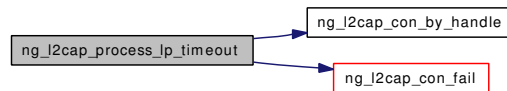
7.56.1.12 void ng_l2cap_process_lp_timeout (node_p node, hook_p hook, void * arg1, int con_handle)

Definition at line 810 of file `ng_l2cap_llpi.c`.

References `ng_l2cap_con::flags`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_handle()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_LP_TIMO`, `NG_L2CAP_TIMEOUT`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, and `ng_l2cap_con::state`.

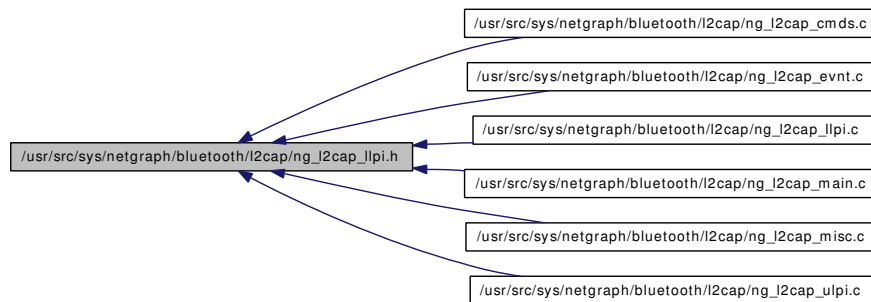
Referenced by `ng_l2cap_lp_timeout()`.

Here is the call graph for this function:



7.57 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int ng_l2cap_lp_con_req (ng_l2cap_p, bdaddr_p)`
- `int ng_l2cap_lp_con_cfm (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_lp_con_ind (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_lp_discon_ind (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_lp_qos_req (ng_l2cap_p, u_int16_t, ng_l2cap_flow_p)`
- `int ng_l2cap_lp_qos_cfm (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_lp_qos_ind (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_lp_send (ng_l2cap_con_p, u_int16_t, struct mbuf *)`
- `int ng_l2cap_lp_receive (ng_l2cap_p, struct mbuf *)`
- `void ng_l2cap_lp_deliver (ng_l2cap_con_p)`
- `void ng_l2cap_process_lp_timeout (node_p, hook_p, void *, int)`
- `void ng_l2cap_process_discon_timeout (node_p, hook_p, void *, int)`

7.57.1 Function Documentation

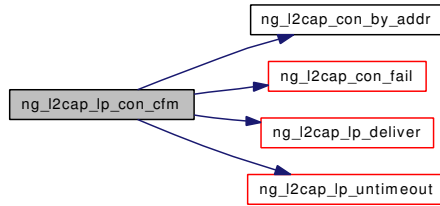
7.57.1.1 `int ng_l2cap_lp_con_cfm (ng_l2cap_p, struct ng_mesg *)`

Definition at line 135 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `ng_l2cap_lp_deliver()`, `ng_l2cap_lp_untimeout()`, `NG_L2CAP_W4_LP_CON_CFM`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



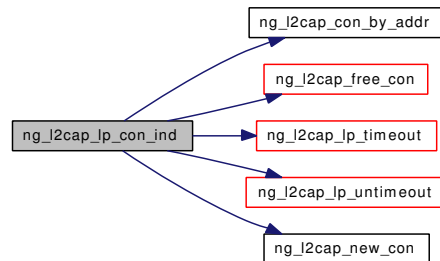
7.57.1.2 int ng_l2cap_lp_con_ind (ng_l2cap_p, struct ng_mesg *)

Definition at line 203 of file ng_l2cap_llpi.c.

References ng_mesg::ng_msghdr::arglen, ng_l2cap::bdaddr, ng_l2cap_con::con_handle, ng_mesg::data, ng_l2cap::hci, ng_mesg::header, NG_HCI_LINK_ACL, NG_HOOK_NOT_VALID, NG_L2CAP_ALERT, ng_l2cap_con_by_addr(), NG_L2CAP_ERR, ng_l2cap_free_con(), NG_L2CAP_HOOK_HCI, ng_l2cap_lp_timeout(), ng_l2cap_lp_untimeout(), ng_l2cap_new_con(), NG_L2CAP_W4_LP_CON_CFM, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_CON_RSP, ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_lower_rcvmsg().

Here is the call graph for this function:



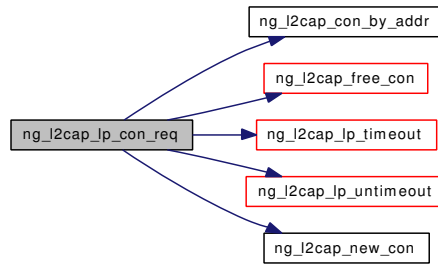
7.57.1.3 int ng_l2cap_lp_con_req (ng_l2cap_p, bdaddr_p)

Definition at line 67 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, ng_mesg::data, ng_l2cap_con::flags, ng_l2cap::hci, NG_HCI_LINK_ACL, NG_HOOK_NOT_VALID, NG_L2CAP_ALERT, ng_l2cap_con_by_addr(), NG_L2CAP_CON_OUTGOING, NG_L2CAP_ERR, ng_l2cap_free_con(), NG_L2CAP_HOOK_HCI, ng_l2cap_lp_timeout(), ng_l2cap_lp_untimeout(), ng_l2cap_new_con(), NG_L2CAP_W4_LP_CON_CFM, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_CON_REQ, ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_get_info_req(), and ng_l2cap_l2ca_ping_req().

Here is the call graph for this function:



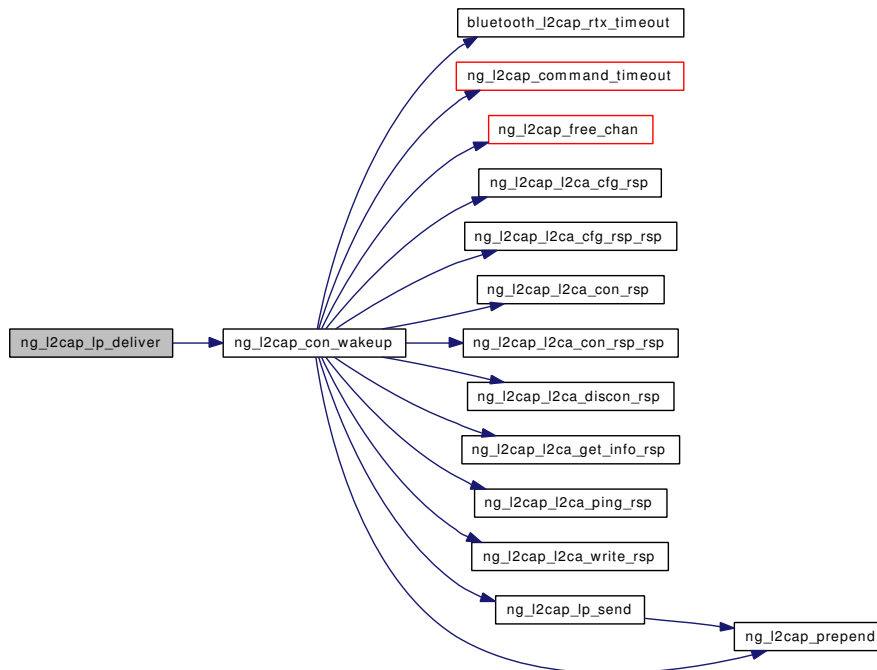
7.57.1.4 void ng_l2cap_ip_deliver (ng_l2cap_con_p)

Definition at line 739 of file `ng_l2cap_llpi.c`.

References `ng_l2cap::hci`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CON_OPEN`, `ng_l2cap_con_wakeup()`, `NG_L2CAP_ERR`, `NG_L2CAP_HOOK_HCI`, `NG_L2CAP_INFO`, `NG_NODE_NAME`, `NG_SEND_DATA_ONLY`, `ng_l2cap::node`, `ng_l2cap_con::state`, and `ng_l2cap_con::tx_pkt`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_ip_con_cfm()`, `ng_l2cap_process_discon_req()`, `ng_l2cap_process_echo_req()`, `ng_l2cap_process_info_req()`, `send_l2cap_cfg_rsp()`, `send_l2cap_con_rej()`, and `send_l2cap_reject()`.

Here is the call graph for this function:



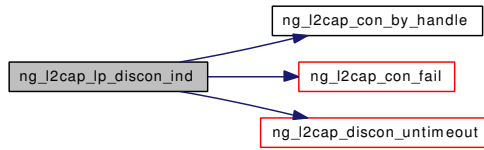
7.57.1.5 `int ng_l2cap_lp_discon_ind (ng_l2cap_p, struct ng_mesg *)`

Definition at line 284 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap_con::flags`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `ng_l2cap_con_by_handle()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_OPEN`, `ng_l2cap_discon_untimeout()`, `NG_L2CAP_ERR`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



7.57.1.6 `int ng_l2cap_lp_qos_cfm (ng_l2cap_p, struct ng_mesg *)`

Definition at line 408 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `NG_NODE_NAME`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

7.57.1.7 `int ng_l2cap_lp_qos_ind (ng_l2cap_p, struct ng_mesg *)`

Definition at line 435 of file `ng_l2cap_llpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_handle()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lower_rcvmsg()`.

Here is the call graph for this function:



7.57.1.8 `int ng_l2cap_lp_qos_req (ng_l2cap_p, u_int16_t, ng_l2cap_flow_p)`

Definition at line 344 of file `ng_l2cap_llpi.c`.

References `ng_l2cap_con::con_handle`, `ng_mesg::data`, `ng_l2cap::hci`, `NG_HOOK_NOT_VALID`, `ng_l2cap_con_by_handle()`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_ERR`, `NG_L2CAP_HOOK_HCI`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_HCI_COOKIE`, `NGM_HCI_LP_QOS_REQ`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Here is the call graph for this function:



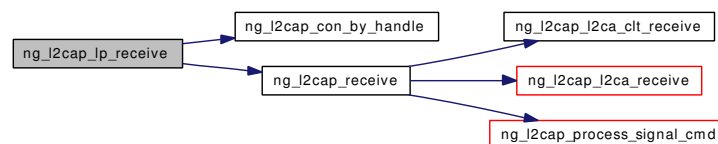
7.57.1.9 int ng_l2cap_ip_receive (ng_l2cap_p, struct mbuf *)

Definition at line 601 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, NG_FREE_M, NG_HCI_CON_HANDLE, NG_HCI_PACKET_FRAGMENT, NG_HCI_PACKET_START, NG_HCI_PB_FLAG, NG_L2CAP_ALERT, ng_l2cap_con_by_handle(), NG_L2CAP_CON_OPEN, NG_L2CAP_ERR, NG_L2CAP_INFO, NG_L2CAP_M_PULLUP, ng_l2cap_receive(), NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, ng_l2cap_con::rx_pkt_len, and ng_l2cap_con::state.

Referenced by ng_l2cap_rcvdata().

Here is the call graph for this function:



7.57.1.10 int ng_l2cap_ip_send (ng_l2cap_con_p, u_int16_t, struct mbuf *)

Definition at line 485 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, ng_l2cap_con::l2cap, NG_FREE_M, NG_HCI_ACL_DATA_PKT, NG_HCI_MK_CON_HANDLE, NG_HCI_PACKET_FRAGMENT, NG_HCI_PACKET_START, NG_L2CAP_ALERT, NG_L2CAP_INFO, ng_l2cap_prepend(), NG_NODE_NAME, ng_l2cap::node, ng_l2cap::pkt_size, and ng_l2cap_con::tx_pkt.

Referenced by ng_l2cap_con_wakeup().

Here is the call graph for this function:



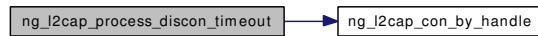
7.57.1.11 void ng_l2cap_process_discon_timeout (node_p, hook_p, void *, int)

Definition at line 853 of file ng_l2cap_llpi.c.

References ng_l2cap_con::con_handle, ng_mesg::data, ng_l2cap_con::flags, ng_l2cap::hci, NG_HOOK_NOT_VALID, NG_L2CAP_ALERT, NG_L2CAP_CON_AUTO_DISCON_TIMO, ng_l2cap_con_by_handle(), NG_L2CAP_ERR, NG_L2CAP_HOOK_HCI, NG_MKMESSAGE, NG_NODE_NAME, NG_NODE_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_MSG_HOOK, NGM_HCI_COOKIE, NGM_HCI_LP_DISCON_REQ, ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_discon_timeout().

Here is the call graph for this function:



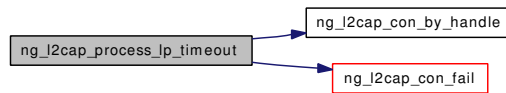
7.57.1.12 void ng_l2cap_process_lp_timeout (node_p, hook_p, void *, int)

Definition at line 810 of file `ng_l2cap_llpi.c`.

References `ng_l2cap_con::flags`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_handle()`, `ng_l2cap_con_fail()`, `NG_L2CAP_CON_LP_TIMO`, `NG_L2CAP_TIMEOUT`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lp_timeout()`.

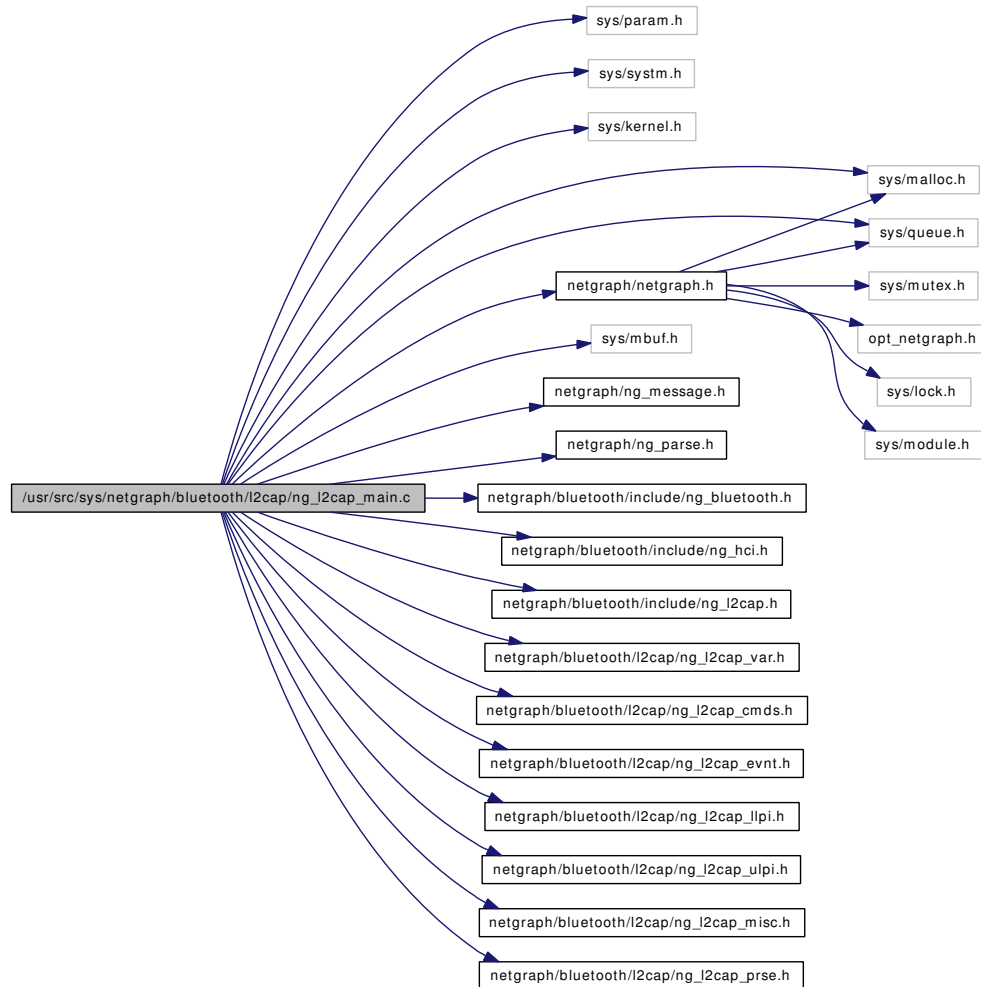
Here is the call graph for this function:



7.58 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_main.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_prse.h>
```

Include dependency graph for ng_l2cap_main.c:



Defines

- #define [M_NETGRAPH_L2CAP](#) M_NETGRAPH

Functions

- [NETGRAPH_INIT](#) (l2cap,&typestruct)
- [MODULE_VERSION](#) (ng_l2cap, NG_BLUETOOTH_VERSION)
- [MODULE_DEPEND](#) (ng_l2cap, ng_bluetooth, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION, NG_BLUETOOTH_VERSION)
- static void [ng_l2cap_cleanup](#) (ng_l2cap_p)
- static void [ng_l2cap_destroy_channels](#) (ng_l2cap_p)
- static int [ng_l2cap_constructor](#) (node_p node)
- static int [ng_l2cap_shutdown](#) (node_p node)
- static int [ng_l2cap_newhook](#) (node_p node, hook_p hook, char const *name)
- static int [ng_l2cap_connect](#) (hook_p hook)
- static int [ng_l2cap_disconnect](#) (hook_p hook)
- static int [ng_l2cap_lower_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)

- static int [ng_l2cap_upper_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_l2cap_default_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_l2cap_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)

Variables

- static [ng_constructor_t](#) [ng_l2cap_constructor](#)
- static [ng_shutdown_t](#) [ng_l2cap_shutdown](#)
- static [ng_newhook_t](#) [ng_l2cap_newhook](#)
- static [ng_connect_t](#) [ng_l2cap_connect](#)
- static [ng_disconnect_t](#) [ng_l2cap_disconnect](#)
- static [ng_rcvmsg_t](#) [ng_l2cap_lower_rcvmsg](#)
- static [ng_rcvmsg_t](#) [ng_l2cap_upper_rcvmsg](#)
- static [ng_rcvmsg_t](#) [ng_l2cap_default_rcvmsg](#)
- static [ng_rcvdata_t](#) [ng_l2cap_rcvdata](#)
- static struct [ng_type](#) [typestruct](#)

7.58.1 Define Documentation

7.58.1.1 #define M_NETGRAPH_L2CAP M_NETGRAPH

Definition at line 65 of file [ng_l2cap_main.c](#).

Referenced by [ng_l2cap_constructor\(\)](#), [ng_l2cap_free_chan\(\)](#), [ng_l2cap_free_con\(\)](#), [ng_l2cap_new_chan\(\)](#), [ng_l2cap_new_cmd\(\)](#), [ng_l2cap_new_con\(\)](#), and [ng_l2cap_shutdown\(\)](#).

7.58.2 Function Documentation

7.58.2.1 MODULE_DEPEND ([ng_l2cap](#), [ng_bluetooth](#), [NG_BLUETOOTH_VERSION](#), [NG_BLUETOOTH_VERSION](#), [NG_BLUETOOTH_VERSION](#))

7.58.2.2 MODULE_VERSION ([ng_l2cap](#), [NG_BLUETOOTH_VERSION](#))

7.58.2.3 NETGRAPH_INIT ([l2cap](#), & [typestruct](#))

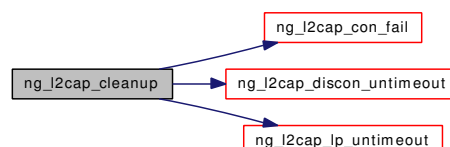
7.58.2.4 static void [ng_l2cap_cleanup](#) ([ng_l2cap_p](#)) [static]

Definition at line 731 of file [ng_l2cap_main.c](#).

References [ng_l2cap_con::flags](#), [NG_L2CAP_CON_AUTO_DISCON_TIMO](#), [ng_l2cap_con_fail\(\)](#), [NG_L2CAP_CON_LP_TIMO](#), [ng_l2cap_discon_untimeout\(\)](#), and [ng_l2cap_lp_untimeout\(\)](#).

Referenced by [ng_l2cap_disconnect\(\)](#), and [ng_l2cap_shutdown\(\)](#).

Here is the call graph for this function:



7.58.2.5 static int ng_l2cap_connect ([hook_p hook](#)) [static]

Definition at line 194 of file ng_l2cap_main.c.

References [ng_l2cap::ctl](#), [ng_l2cap::hci](#), [ng_l2cap::l2c](#), [NG_HOOK_NODE](#), [NG_HOOK_SET_-RCVMSG](#), [ng_l2cap_lower_rcvmsg](#), [ng_l2cap_send_hook_info\(\)](#), [ng_l2cap_upper_rcvmsg](#), [NG_-NODE_PRIVATE](#), [ng_send_fn](#), and [ng_l2cap::node](#).

Here is the call graph for this function:



7.58.2.6 static int ng_l2cap_constructor ([node_p node](#)) [static]

Definition at line 111 of file ng_l2cap_main.c.

References [M_NETGRAPH_L2CAP](#), [NG_L2CAP_WARN_LEVEL](#), [NG_NODE_FORCE_WRITER](#), and [NG_NODE_SET_PRIVATE](#).

7.58.2.7 static int ng_l2cap_default_rcvmsg ([node_p node](#), [item_p item](#), [hook_p lasthook](#)) [static]

Definition at line 473 of file ng_l2cap_main.c.

References [ng_mesg::ng_msghdr::arglen](#), [ng_l2cap::bdaddr](#), [ng_mesg::ng_msghdr::cmd](#), [ng_l2cap_chan::con](#), [ng_l2cap_con::con_handle](#), [ng_l2cap::ctl](#), [ng_mesg::data](#), [ng_l2cap_chan::dcid](#), [ng_l2cap_node_chan_ep::dcid](#), [ng_l2cap::debug](#), [ng_l2cap::discon_timo](#), [ng_l2cap_con::flags](#), [ng_l2cap::flags](#), [ng_l2cap::hci](#), [ng_mesg::header](#), [ng_l2cap_chan::imtu](#), [ng_l2cap_node_chan_ep::imtu](#), [ng_l2cap::l2c](#), [NG_FREE_MSG](#), [NG_L2CAP_CON_RX](#), [NG_L2CAP_CON_TX](#), [NG_L2CAP_HOOK_CTL](#), [NG_-L2CAP_HOOK_HCI](#), [NG_L2CAP_HOOK_L2C](#), [NG_L2CAP_MAX_CHAN_NUM](#), [NG_L2CAP_-MAX_CON_NUM](#), [NG_MKRESPONSE](#), [NG_NODE_PRIVATE](#), [NG_RESPOND_MSG](#), [NG_-TEXTRESPONSE](#), [NGI_GET_MSG](#), [NGM_GENERIC_COOKIE](#), [NGM_L2CAP_COOKIE](#), [NGM_-L2CAP_NODE_GET_AUTO_DISCON_TIMO](#), [NGM_L2CAP_NODE_GET_CHAN_LIST](#), [NGM_-L2CAP_NODE_GET_CON_LIST](#), [NGM_L2CAP_NODE_GET_DEBUG](#), [NGM_L2CAP_NODE_-GET_FLAGS](#), [NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO](#), [NGM_L2CAP_NODE_-SET_DEBUG](#), [NGM_TEXT_STATUS](#), [ng_l2cap_node_chan_list_ep::num_channels](#), [ng_l2cap_chan::omtu](#), [ng_l2cap_node_chan_ep::omtu](#), [ng_l2cap_con::pending](#), [ng_l2cap::pkt_size](#), [ng_l2cap_chan::psm](#), [ng_l2cap_node_chan_ep::psm](#), [ng_l2cap_node_chan_ep::remote](#), [ng_l2cap_con::remote](#), [ng_l2cap_con::rx_pkt](#), [ng_l2cap_chan::scid](#), [ng_l2cap_node_chan_ep::scid](#), [ng_l2cap_chan::state](#), [ng_l2cap_node_-chan_ep::state](#), [ng_l2cap_con::state](#), [ng_l2cap_con::tx_pkt](#), and [ng_mesg::ng_msghdr::typecookie](#).

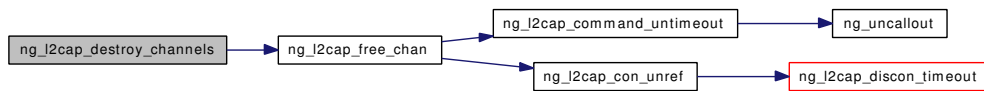
7.58.2.8 static void ng_l2cap_destroy_channels ([ng_l2cap_p](#)) [static]

Definition at line 754 of file ng_l2cap_main.c.

References [ng_l2cap_free_chan\(\)](#).

Referenced by [ng_l2cap_disconnect\(\)](#).

Here is the call graph for this function:

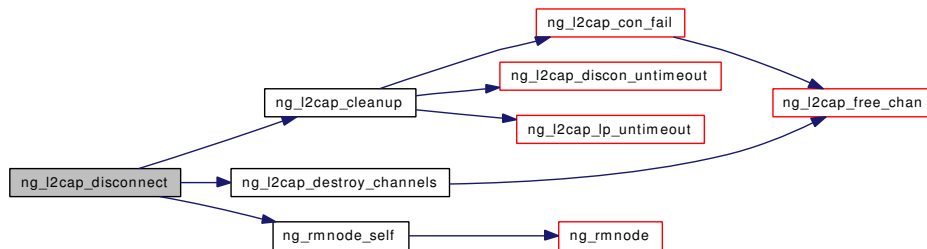


7.58.2.9 static int ng_l2cap_disconnect ([hook_p hook](#)) [static]

Definition at line 225 of file ng_l2cap_main.c.

References [ng_l2cap::ctl](#), [ng_l2cap::hci](#), [ng_l2cap::l2c](#), [NG_HOOK_NODE](#), [ng_l2cap_cleanup\(\)](#), [ng_l2cap_destroy_channels\(\)](#), [NG_NODE_IS_VALID](#), [NG_NODE_NUMHOOKS](#), [NG_NODE_PRIVATE](#), and [ng_rmnode_self\(\)](#).

Here is the call graph for this function:

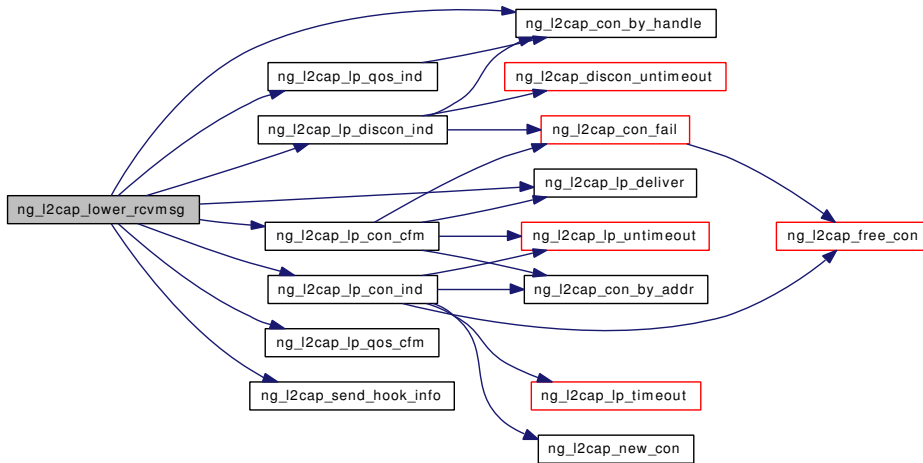


7.58.2.10 static int ng_l2cap_lower_rcvmsg ([node_p node](#), [item_p item](#), [hook_p lasthook](#)) [static]

Definition at line 258 of file ng_l2cap_main.c.

References [ng_mesg::ng_msghdr::arglen](#), [ng_l2cap::bdaddr](#), [ng_mesg::ng_msghdr::cmd](#), [ng_l2cap::con::con_handle](#), [ng_l2cap::ctl](#), [ng_mesg::data](#), [ng_mesg::header](#), [ng_l2cap::l2c](#), [NG_FREE_ITEM](#), [ng_l2cap_con_by_handle\(\)](#), [ng_l2cap_default_rcvmsg](#), [NG_L2CAP_INFO](#), [ng_l2cap_lp_con_cfm\(\)](#), [ng_l2cap_lp_con_ind\(\)](#), [ng_l2cap_lp_deliver\(\)](#), [ng_l2cap_lp_discon_ind\(\)](#), [ng_l2cap_lp_qos_cfm\(\)](#), [ng_l2cap_lp_qos_ind\(\)](#), [ng_l2cap_send_hook_info\(\)](#), [NG_L2CAP_WARN](#), [NG_NODE_NAME](#), [NG_NODE_PRIVATE](#), [NGI_MSG](#), [NGM_HCI_COOKIE](#), [NGM_HCI_LP_CON_CFM](#), [NGM_HCI_LP_CON_IND](#), [NGM_HCI_LP_DISCON_IND](#), [NGM_HCI_LP_QOS_CFM](#), [NGM_HCI_LP_QOS_IND](#), [NGM_HCI_NODE_UP](#), [NGM_HCI_SYNC_CON_QUEUE](#), [ng_l2cap::node](#), [ng_l2cap::num_pkts](#), [ng_l2cap_con::pending](#), [ng_l2cap::pkt_size](#), and [ng_mesg::ng_msghdr::typecookie](#).

Here is the call graph for this function:



7.58.2.11 static int ng_l2cap_newhook (node_p node, hook_p hook, char const * name) [static]

Definition at line 167 of file ng_l2cap_main.c.

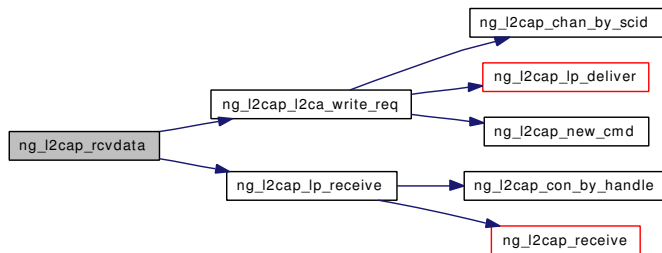
References ng_l2cap::ctl, ng_l2cap::hci, ng_l2cap::l2c, NG_L2CAP_HOOK_CTL, NG_L2CAP_HOOK_HCI, NG_L2CAP_HOOK_L2C, and NG_NODE_PRIVATE.

7.58.2.12 static int ng_l2cap_rcvdata (hook_p hook, item_p item) [static]

Definition at line 704 of file ng_l2cap_main.c.

References ng_l2cap::hci, ng_l2cap::l2c, NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NODE, ng_l2cap_l2ca_write_req(), ng_l2cap_ip_receive(), NG_NODE_PRIVATE, and NGI_GET_M.

Here is the call graph for this function:

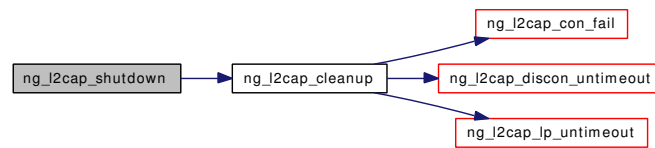


7.58.2.13 static int ng_l2cap_shutdown (node_p node) [static]

Definition at line 139 of file ng_l2cap_main.c.

References M_NETGRAPH_L2CAP, ng_l2cap_cleanup(), NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_l2cap::node.

Here is the call graph for this function:

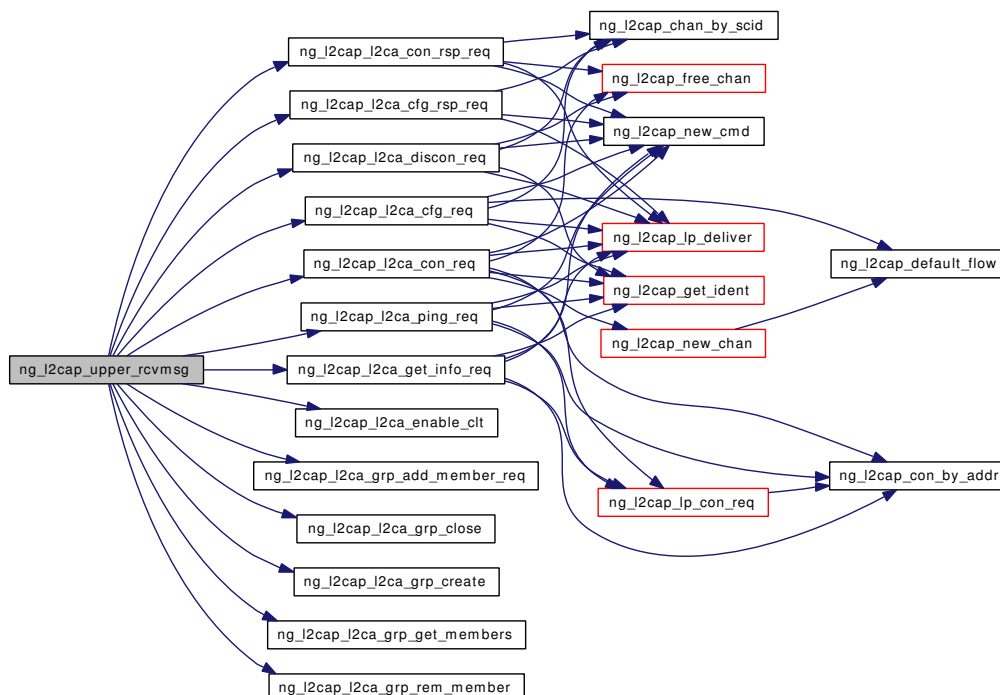


7.58.2.14 static int ng_l2cap_upper_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 378 of file ng_l2cap_main.c.

References ng_mesg::ng_msghdr::cmd, ng_mesg::header, NG_FREE_ITEM, ng_l2cap_default_rcvmsg, ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_grp_add_member_req(), ng_l2cap_l2ca_grp_close(), ng_l2cap_l2ca_grp_create(), ng_l2cap_l2ca_grp_get_members(), ng_l2cap_l2ca_grp_rem_member(), ng_l2cap_l2ca_ping_req(), NG_NODE_PRIVATE, NGI_MSG, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_CFG, NGM_L2CAP_L2CA_CFG_RSP, NGM_L2CAP_L2CA_CON, NGM_L2CAP_L2CA_CON_RSP, NGM_L2CAP_L2CA_DISCON, NGM_L2CAP_L2CA_ENABLE_CLT, NGM_L2CAP_L2CA_GET_INFO, NGM_L2CAP_L2CA_GRP_ADD_MEMBER, NGM_L2CAP_L2CA_GRP_CLOSE, NGM_L2CAP_L2CA_GRP_CREATE, NGM_L2CAP_L2CA_GRP_MEMBERSHIP, NGM_L2CAP_L2CA_GRP_REM_MEMBER, NGM_L2CAP_L2CA_PING, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



7.58.3 Variable Documentation

7.58.3.1 `ng_connect_t ng_l2cap_connect` [static]

Definition at line 72 of file `ng_l2cap_main.c`.

7.58.3.2 `ng_constructor_t ng_l2cap_constructor` [static]

Definition at line 69 of file `ng_l2cap_main.c`.

7.58.3.3 `ng_rcvmsg_t ng_l2cap_default_rcvmsg` [static]

Definition at line 76 of file `ng_l2cap_main.c`.

Referenced by `ng_l2cap_lower_rcvmsg()`, and `ng_l2cap_upper_rcvmsg()`.

7.58.3.4 `ng_disconnect_t ng_l2cap_disconnect` [static]

Definition at line 73 of file `ng_l2cap_main.c`.

7.58.3.5 `ng_rcvmsg_t ng_l2cap_lower_rcvmsg` [static]

Definition at line 74 of file `ng_l2cap_main.c`.

Referenced by `ng_l2cap_connect()`.

7.58.3.6 `ng_newhook_t ng_l2cap_newhook` [static]

Definition at line 71 of file `ng_l2cap_main.c`.

7.58.3.7 `ng_rcvdata_t ng_l2cap_rcvdata` [static]

Definition at line 77 of file `ng_l2cap_main.c`.

7.58.3.8 `ng_shutdown_t ng_l2cap_shutdown` [static]

Definition at line 70 of file `ng_l2cap_main.c`.

7.58.3.9 `ng_rcvmsg_t ng_l2cap_upper_rcvmsg` [static]

Definition at line 75 of file `ng_l2cap_main.c`.

Referenced by `ng_l2cap_connect()`.

7.58.3.10 `struct ng_type typestruct` [static]

Initial value:

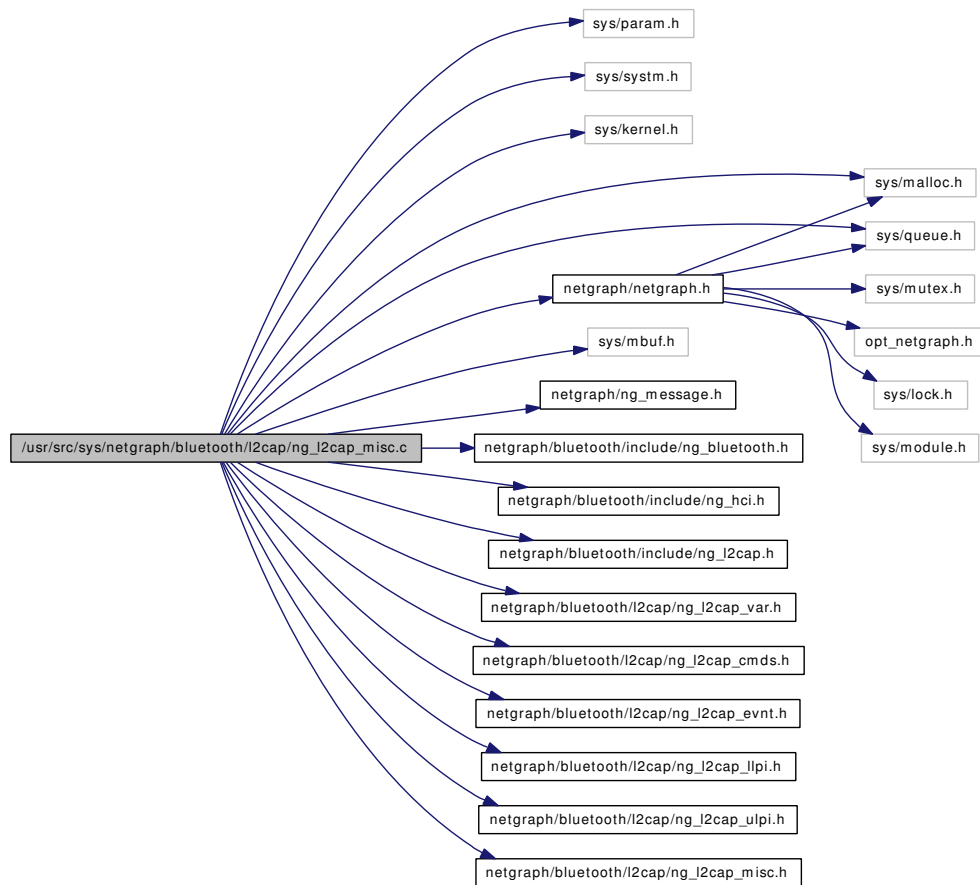

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_L2CAP_NODE_TYPE,
    .constructor = ng_l2cap_constructor,
    .rcvmsg =       ng_l2cap_default_rcvmsg,
    .shutdown =     ng_l2cap_shutdown,
    .newhook =      ng_l2cap_newhook,
    .connect =      ng_l2cap_connect,
    .rcvdata =      ng_l2cap_rcvdata,
    .disconnect =   ng_l2cap_disconnect,
    .cmdlist =      ng_l2cap_cmdlist,
}
```

Definition at line 80 of file ng_l2cap_main.c.

7.59 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
```

Include dependency graph for ng_l2cap_misc.c:



Functions

- static u_int16_t [ng_l2cap_get_cid](#) (ng_l2cap_p)
- void [ng_l2cap_send_hook_info](#) (node_p node, hook_p hook, void *arg1, int arg2)
- ng_l2cap_con_p [ng_l2cap_new_con](#) (ng_l2cap_p l2cap, bdaddr_p bdaddr)
- void [ng_l2cap_con_ref](#) (ng_l2cap_con_p con)
- void [ng_l2cap_con_unref](#) (ng_l2cap_con_p con)
- int [ng_l2cap_discon_timeout](#) (ng_l2cap_con_p con)
- int [ng_l2cap_discon_untimeout](#) (ng_l2cap_con_p con)
- void [ng_l2cap_free_con](#) (ng_l2cap_con_p con)
- ng_l2cap_con_p [ng_l2cap_con_by_addr](#) (ng_l2cap_p l2cap, bdaddr_p bdaddr)
- ng_l2cap_con_p [ng_l2cap_con_by_handle](#) (ng_l2cap_p l2cap, u_int16_t con_handle)
- ng_l2cap_chan_p [ng_l2cap_new_chan](#) (ng_l2cap_p l2cap, ng_l2cap_con_p con, u_int16_t psm)
- ng_l2cap_chan_p [ng_l2cap_chan_by_scid](#) (ng_l2cap_p l2cap, u_int16_t scid)
- void [ng_l2cap_free_chan](#) (ng_l2cap_chan_p ch)
- ng_l2cap_cmd_p [ng_l2cap_new_cmd](#) (ng_l2cap_con_p con, ng_l2cap_chan_p ch, u_int8_t ident, u_int8_t code, u_int32_t token)
- ng_l2cap_cmd_p [ng_l2cap_cmd_by_ident](#) (ng_l2cap_con_p con, u_int8_t ident)
- int [ng_l2cap_lp_timeout](#) (ng_l2cap_con_p con)
- int [ng_l2cap_lp_untimeout](#) (ng_l2cap_con_p con)
- int [ng_l2cap_command_timeout](#) (ng_l2cap_cmd_p cmd, int timo)
- int [ng_l2cap_command_untimeout](#) (ng_l2cap_cmd_p cmd)

- mbuf * [ng_l2cap_prepend](#) (struct mbuf *m, int size)
- [ng_l2cap_flow_p ng_l2cap_default_flow](#) (void)
- u_int8_t [ng_l2cap_get_ident](#) ([ng_l2cap_con_p con](#))

7.59.1 Function Documentation

7.59.1.1 [ng_l2cap_chan_p ng_l2cap_chan_by_scid](#) ([ng_l2cap_p l2cap](#), [u_int16_t scid](#))

Definition at line 372 of file [ng_l2cap_misc.c](#).

References [ng_l2cap_chan::scid](#).

Referenced by [ng_l2cap_get_cid\(\)](#), [ng_l2cap_l2ca_cfg_req\(\)](#), [ng_l2cap_l2ca_cfg_rsp_req\(\)](#), [ng_l2cap_l2ca_con_rsp_req\(\)](#), [ng_l2cap_l2ca_discon_req\(\)](#), [ng_l2cap_l2ca_receive\(\)](#), [ng_l2cap_l2ca_write_req\(\)](#), [ng_l2cap_process_cfg_req\(\)](#), and [ng_l2cap_process_discon_req\(\)](#).

7.59.1.2 [ng_l2cap_cmd_p ng_l2cap_cmd_by_ident](#) ([ng_l2cap_con_p con](#), [u_int8_t ident](#))

Definition at line 448 of file [ng_l2cap_misc.c](#).

References [ng_l2cap_con::l2cap](#), [NG_L2CAP_CMD_PENDING](#), [NG_NODE_NAME](#), and [ng_l2cap::node](#).

Referenced by [ng_l2cap_get_ident\(\)](#), [ng_l2cap_process_cfg_rsp\(\)](#), [ng_l2cap_process_cmd_rej\(\)](#), [ng_l2cap_process_command_timeout\(\)](#), [ng_l2cap_process_con_rsp\(\)](#), [ng_l2cap_process_discon_rsp\(\)](#), [ng_l2cap_process_echo_rsp\(\)](#), and [ng_l2cap_process_info_rsp\(\)](#).

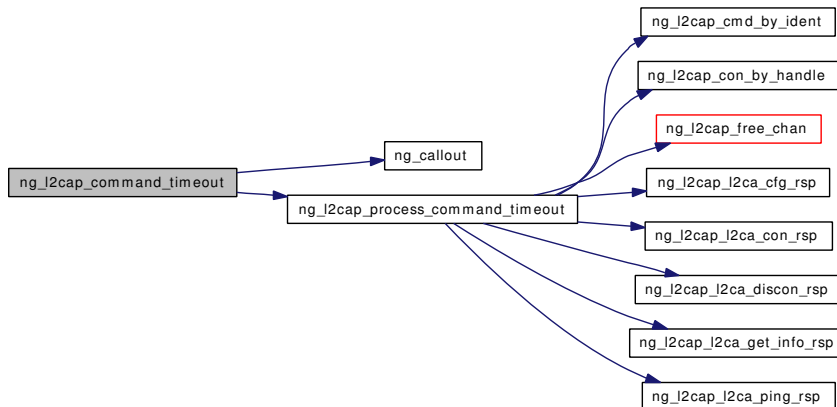
7.59.1.3 [int ng_l2cap_command_timeout](#) ([ng_l2cap_cmd_p cmd](#), [int timo](#))

Definition at line 515 of file [ng_l2cap_misc.c](#).

References [ng_callout\(\)](#), [NG_L2CAP_CMD_PENDING](#), [ng_l2cap_process_command_timeout\(\)](#), and [NG_NODE_NAME](#).

Referenced by [ng_l2cap_con_wakeup\(\)](#), and [ng_l2cap_process_con_rsp\(\)](#).

Here is the call graph for this function:



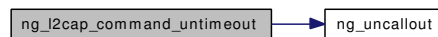
7.59.1.4 `int ng_l2cap_command_untimeout (ng_l2cap_cmd_p cmd)`

Definition at line 538 of file ng_l2cap_misc.c.

References NG_L2CAP_CMD_PENDING, NG_NODE_NAME, and ng_uncallout().

Referenced by ng_l2cap_con_fail(), ng_l2cap_free_chan(), ng_l2cap_free_con(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_echo_rsp(), and ng_l2cap_process_info_rsp().

Here is the call graph for this function:

**7.59.1.5** `ng_l2cap_con_p ng_l2cap_con_by_addr (ng_l2cap_p l2cap, bdaddr_p bdaddr)`

Definition at line 295 of file ng_l2cap_misc.c.

References ng_l2cap_con::remote.

Referenced by ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_ping_req(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), and ng_l2cap_lp_con_req().

7.59.1.6 `ng_l2cap_con_p ng_l2cap_con_by_handle (ng_l2cap_p l2cap, u_int16_t con_handle)`

Definition at line 311 of file ng_l2cap_misc.c.

References ng_l2cap_con::con_handle.

Referenced by ng_l2cap_lower_rcvmsg(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), ng_l2cap_lp_receive(), ng_l2cap_process_command_timeout(), ng_l2cap_process_discon_timeout(), and ng_l2cap_process_lp_timeout().

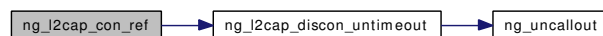
7.59.1.7 `void ng_l2cap_con_ref (ng_l2cap_con_p con)`

Definition at line 150 of file ng_l2cap_misc.c.

References ng_l2cap_con::flags, ng_l2cap_con::l2cap, NG_L2CAP_CON_AUTO_DISCON_TIMO, NG_L2CAP_CON_OPEN, NG_L2CAP_CON_OUTGOING, ng_l2cap_discon_untimeout(), NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::refcnt, and ng_l2cap_con::state.

Referenced by ng_l2cap_new_chan().

Here is the call graph for this function:

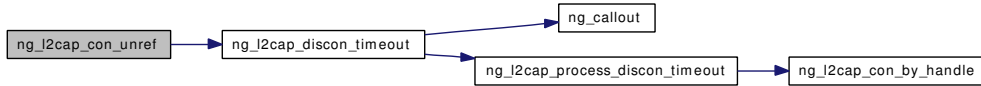
**7.59.1.8** `void ng_l2cap_con_unref (ng_l2cap_con_p con)`

Definition at line 171 of file ng_l2cap_misc.c.

References `ng_l2cap::discon_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_DYING`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_CON_OUTGOING`, `ng_l2cap_discon_timeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_con::refcnt`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_free_chan()`.

Here is the call graph for this function:



7.59.1.9 `ng_l2cap_flow_p ng_l2cap_default_flow (void)`

Definition at line 573 of file `ng_l2cap_misc.c`.

References `NG_HCI_SERVICE_TYPE_BEST_EFFORT`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, and `ng_l2cap_new_chan()`.

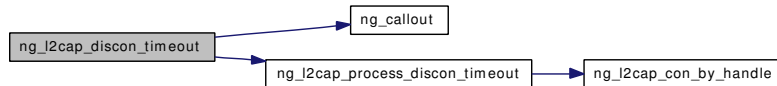
7.59.1.10 `int ng_l2cap_discon_timeout (ng_l2cap_con_p con)`

Definition at line 202 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::con_handle`, `ng_l2cap_con::con_timo`, `ng_l2cap::discon_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `ng_callout()`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_L2CAP_CON_LP_TIMO`, `ng_l2cap_process_discon_timeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_con_unref()`.

Here is the call graph for this function:



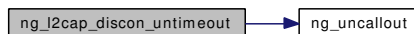
7.59.1.11 `int ng_l2cap_discon_untimeout (ng_l2cap_con_p con)`

Definition at line 224 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::con_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_NODE_NAME`, `ng_uncallout()`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_cleanup()`, `ng_l2cap_con_ref()`, and `ng_l2cap_lp_discon_ind()`.

Here is the call graph for this function:



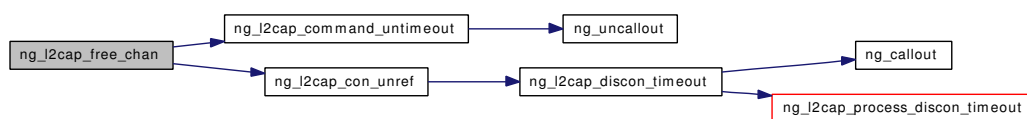
7.59.1.12 void ng_l2cap_free_chan (ng_l2cap_chan_p ch)

Definition at line 388 of file ng_l2cap_misc.c.

References ng_l2cap_cmd::ch, ng_l2cap_chan::con, M_NETGRAPH_L2CAP, NG_L2CAP_CMD_PENDING, ng_l2cap_command_untimeout(), ng_l2cap_con_unref(), ng_l2cap_free_cmd, and ng_l2cap_unlink_cmd.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_destroy_channels(), ng_l2cap_free_con(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), and ng_l2cap_process_discon_rsp().

Here is the call graph for this function:

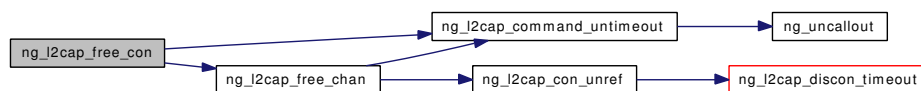
**7.59.1.13 void ng_l2cap_free_con (ng_l2cap_con_p con)**

Definition at line 245 of file ng_l2cap_misc.c.

References ng_l2cap_chan::con, ng_l2cap_con::flags, ng_l2cap_con::l2cap, M_NETGRAPH_L2CAP, NG_FREE_M, NG_L2CAP_CMD_PENDING, ng_l2cap_command_untimeout(), NG_L2CAP_CON_AUTO_DISCON_TIMO, NG_L2CAP_CON_CLOSED, NG_L2CAP_CON_LP_TIMO, ng_l2cap_free_chan(), ng_l2cap_free_cmd, ng_l2cap_unlink_cmd, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_con::rx_pkt, ng_l2cap_con::state, and ng_l2cap_con::tx_pkt.

Referenced by ng_l2cap_con_fail(), ng_l2cap_lp_con_ind(), and ng_l2cap_lp_con_req().

Here is the call graph for this function:

**7.59.1.14 static u_int16_t ng_l2cap_get_cid (ng_l2cap_p) [static]**

Definition at line 594 of file ng_l2cap_misc.c.

References ng_l2cap::cid, ng_l2cap_chan_by_scid(), NG_L2CAP_FIRST_CID, and NG_L2CAP_NULL_CID.

Referenced by ng_l2cap_new_chan().

Here is the call graph for this function:



7.59.1.15 `u_int8_t ng_l2cap_get_ident (ng_l2cap_con_p con)`

Definition at line 622 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::ident`, `ng_l2cap_cmd_by_ident()`, `NG_L2CAP_FIRST_IDENT`, and `NG_L2CAP_NULL_IDENT`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, and `ng_l2cap_l2ca_ping_req()`.

Here is the call graph for this function:



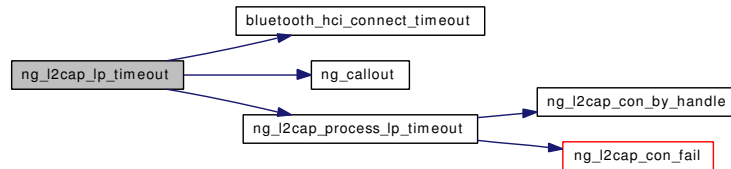
7.59.1.16 `int ng_l2cap_lp_timeout (ng_l2cap_con_p con)`

Definition at line 471 of file `ng_l2cap_misc.c`.

References `bluetooth_hci_connect_timeout()`, `ng_l2cap_con::con_handle`, `ng_l2cap_con::con_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `ng_callout()`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_L2CAP_CON_LP_TIMO`, `ng_l2cap_process_lp_timeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

Here is the call graph for this function:



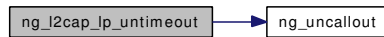
7.59.1.17 `int ng_l2cap_lp_untimeout (ng_l2cap_con_p con)`

Definition at line 493 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::con_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_LP_TIMO`, `NG_NODE_NAME`, `ng_unccallout()`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_cleanup()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

Here is the call graph for this function:



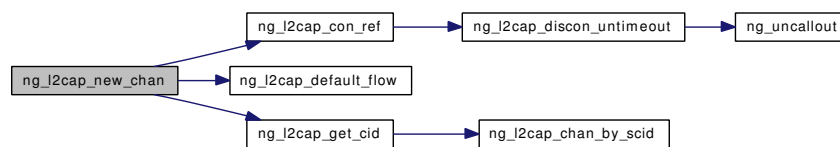
7.59.1.18 `ng_l2cap_chan_p` `ng_l2cap_new_chan` (`ng_l2cap_p` `l2cap`, `ng_l2cap_con_p` `con`, `u_int16_t` `psm`)

Definition at line 328 of file `ng_l2cap_misc.c`.

References `M_NETGRAPH_L2CAP`, `NG_L2CAP_CLOSED`, `ng_l2cap_con_ref()`, `ng_l2cap_default_flow()`, `NG_L2CAP_FLUSH_TIMO_DEFAULT`, `ng_l2cap_get_cid()`, `NG_L2CAP_LINK_TIMO_DEFAULT`, `NG_L2CAP_MTU_DEFAULT`, `NG_L2CAP_NULL_CID`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_l2ca_con_req()`, and `ng_l2cap_process_con_req()`.

Here is the call graph for this function:



7.59.1.19 `ng_l2cap_cmd_p` `ng_l2cap_new_cmd` (`ng_l2cap_con_p` `con`, `ng_l2cap_chan_p` `ch`, `u_int8_t` `ident`, `u_int8_t` `code`, `u_int32_t` `token`)

Definition at line 419 of file `ng_l2cap_misc.c`.

References `ng_l2cap_chan::con`, `ng_l2cap_chan::ident`, `ng_l2cap_con::l2cap`, `M_NETGRAPH_L2CAP`, `ng_callout_init`, `NG_NODE_NAME`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_process_discon_req()`, `ng_l2cap_process_echo_req()`, `ng_l2cap_process_info_req()`, `send_l2cap_cfg_rsp()`, `send_l2cap_con_rej()`, and `send_l2cap_reject()`.

7.59.1.20 `ng_l2cap_con_p` `ng_l2cap_new_con` (`ng_l2cap_p` `l2cap`, `bdaddr_p` `bdaddr`)

Definition at line 101 of file `ng_l2cap_misc.c`.

References `M_NETGRAPH_L2CAP`, `ng_callout_init`, `NG_L2CAP_CON_CLOSED`, and `NG_L2CAP_FIRST_IDENT`.

Referenced by `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

7.59.1.21 `struct mbuf*` `ng_l2cap_prepend` (`struct mbuf*` `m`, `int` `size`)

Definition at line 559 of file `ng_l2cap_misc.c`.

Referenced by `ng_l2cap_con_wakeup()`, `ng_l2cap_lp_send()`, and `ng_l2cap_process_echo_req()`.

7.59.1.22 `void` `ng_l2cap_send_hook_info` (`node_p` `node`, `hook_p` `hook`, `void*` `arg1`, `int` `arg2`)

Definition at line 65 of file `ng_l2cap_misc.c`.

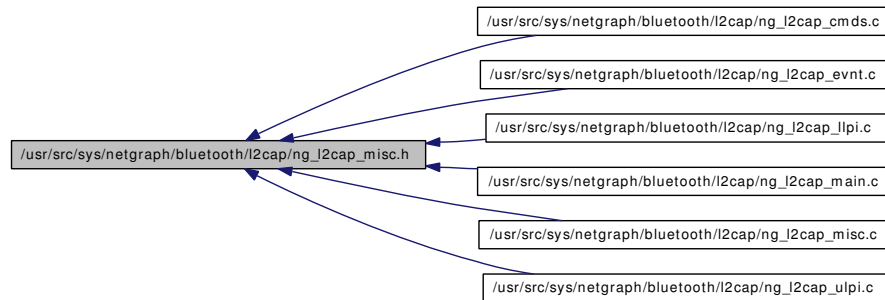
References `ng_l2cap::bdaddr`, `ng_mesg::data`, `ng_l2cap::hci`, `NG_HCI_BDADDR_ANY`, `NG_HOOK_NAME`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_INFO`, `NG_MKMESSAGE`, `NG_NODE_NAME`,

NG_NODE_NOT_VALID, NG_NODE_PRIVATE, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_NODE_HOOK_INFO, and ng_l2cap::node.

Referenced by ng_l2cap_connect(), and ng_l2cap_lower_rcvmsg().

7.60 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [ng_l2cap_link_cmd](#)(con, cmd)
- #define [ng_l2cap_unlink_cmd](#)(cmd)
- #define [ng_l2cap_free_cmd](#)(cmd)

Functions

- void [ng_l2cap_send_hook_info](#) (node_p, hook_p, void *, int)
- [ng_l2cap_con_p ng_l2cap_new_con](#) (ng_l2cap_p, bdaddr_p)
- void [ng_l2cap_con_ref](#) (ng_l2cap_con_p)
- void [ng_l2cap_con_unref](#) (ng_l2cap_con_p)
- [ng_l2cap_con_p ng_l2cap_con_by_addr](#) (ng_l2cap_p, bdaddr_p)
- [ng_l2cap_con_p ng_l2cap_con_by_handle](#) (ng_l2cap_p, u_int16_t)
- void [ng_l2cap_free_con](#) (ng_l2cap_con_p)
- [ng_l2cap_chan_p ng_l2cap_new_chan](#) (ng_l2cap_p, ng_l2cap_con_p, u_int16_t)
- [ng_l2cap_chan_p ng_l2cap_chan_by_scid](#) (ng_l2cap_p, u_int16_t)
- void [ng_l2cap_free_chan](#) (ng_l2cap_chan_p)
- [ng_l2cap_cmd_p ng_l2cap_new_cmd](#) (ng_l2cap_con_p, ng_l2cap_chan_p, u_int8_t, u_int8_t, u_int32_t)
- [ng_l2cap_cmd_p ng_l2cap_cmd_by_ident](#) (ng_l2cap_con_p, u_int8_t)
- u_int8_t [ng_l2cap_get_ident](#) (ng_l2cap_con_p)
- int [ng_l2cap_discon_timeout](#) (ng_l2cap_con_p)
- int [ng_l2cap_discon_untimeout](#) (ng_l2cap_con_p)
- int [ng_l2cap_lp_timeout](#) (ng_l2cap_con_p)
- int [ng_l2cap_lp_untimeout](#) (ng_l2cap_con_p)
- int [ng_l2cap_command_timeout](#) (ng_l2cap_cmd_p, int)
- int [ng_l2cap_command_untimeout](#) (ng_l2cap_cmd_p)
- mbuf * [ng_l2cap_prepend](#) (struct mbuf *, int)
- [ng_l2cap_flow_p ng_l2cap_default_flow](#) (void)

7.60.1 Define Documentation

7.60.1.1 #define ng_l2cap_free_cmd(cmd)

Value:

```
do { \
    KASSERT(!callout_pending(&(cmd)->timo), ("Pending callout!")); \
    NG_FREE_M((cmd)->aux); \
    bzero((cmd), sizeof(*(cmd))); \
    FREE((cmd), M_NETGRAPH_L2CAP); \
} while (0)
```

Definition at line 74 of file ng_l2cap_misc.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_free_chan(), ng_l2cap_free_con(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_ping_req(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), send_l2cap_cfg_rsp(), send_l2cap_con_rej(), and send_l2cap_reject().

7.60.1.2 #define ng_l2cap_link_cmd(con, cmd)

Value:

```
do { \
    TAILQ_INSERT_TAIL(&(con)->cmd_list, (cmd), next); \
    ng_l2cap_con_ref((con)); \
} while (0)
```

Definition at line 62 of file ng_l2cap_misc.h.

Referenced by ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_write_req(), ng_l2cap_process_discon_req(), ng_l2cap_process_echo_req(), ng_l2cap_process_info_req(), send_l2cap_cfg_rsp(), send_l2cap_con_rej(), and send_l2cap_reject().

7.60.1.3 #define ng_l2cap_unlink_cmd(cmd)

Value:

```
do { \
    TAILQ_REMOVE(&((cmd)->con->cmd_list), (cmd), next); \
    ng_l2cap_con_unref((cmd)->con); \
} while (0)
```

Definition at line 68 of file ng_l2cap_misc.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_free_chan(), ng_l2cap_free_con(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), ng_l2cap_process_con_rsp(), ng_l2cap_process_echo_rsp(), and ng_l2cap_process_info_rsp().

7.60.2 Function Documentation

7.60.2.1 [ng_l2cap_chan_p](#) [ng_l2cap_chan_by_scid](#) ([ng_l2cap_p](#), [u_int16_t](#))

Definition at line 372 of file `ng_l2cap_misc.c`.

References `ng_l2cap_chan::scid`.

Referenced by `ng_l2cap_get_cid()`, `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_process_cfg_req()`, and `ng_l2cap_process_discon_req()`.

7.60.2.2 [ng_l2cap_cmd_p](#) [ng_l2cap_cmd_by_ident](#) ([ng_l2cap_con_p](#), [u_int8_t](#))

Definition at line 448 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::l2cap`, `NG_L2CAP_CMD_PENDING`, `NG_NODE_NAME`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_get_ident()`, `ng_l2cap_process_cfg_rsp()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_rsp()`, `ng_l2cap_process_echo_rsp()`, and `ng_l2cap_process_info_rsp()`.

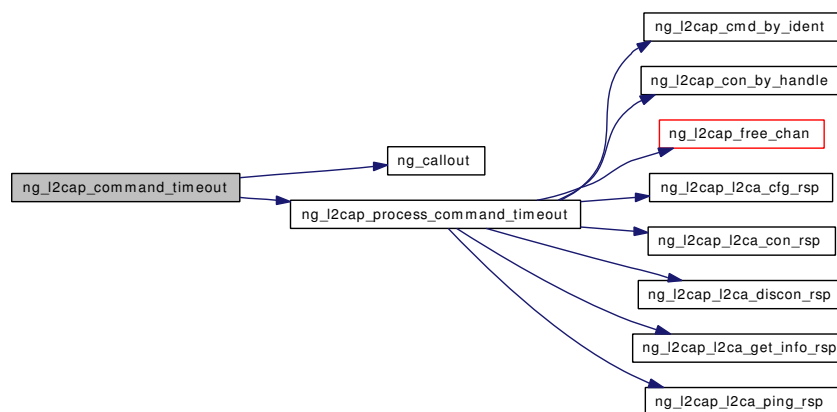
7.60.2.3 [int](#) [ng_l2cap_command_timeout](#) ([ng_l2cap_cmd_p](#), [int](#))

Definition at line 515 of file `ng_l2cap_misc.c`.

References `ng_callout()`, `NG_L2CAP_CMD_PENDING`, `ng_l2cap_process_command_timeout()`, and `NG_NODE_NAME`.

Referenced by `ng_l2cap_con_wakeup()`, and `ng_l2cap_process_con_rsp()`.

Here is the call graph for this function:



7.60.2.4 [int](#) [ng_l2cap_command_untimeout](#) ([ng_l2cap_cmd_p](#))

Definition at line 538 of file `ng_l2cap_misc.c`.

References `NG_L2CAP_CMD_PENDING`, `NG_NODE_NAME`, and `ng_uncallout()`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_free_chan()`, `ng_l2cap_free_con()`, `ng_l2cap_process_cfg_rsp()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_rsp()`, `ng_l2cap_process_echo_rsp()`, and `ng_l2cap_process_info_rsp()`.

Here is the call graph for this function:



7.60.2.5 `ng_l2cap_con_p ng_l2cap_con_by_addr (ng_l2cap_p, bdaddr_p)`

Definition at line 295 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::remote`.

Referenced by `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

7.60.2.6 `ng_l2cap_con_p ng_l2cap_con_by_handle (ng_l2cap_p, u_int16_t)`

Definition at line 311 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::con_handle`.

Referenced by `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_lp_discon_ind()`, `ng_l2cap_lp_qos_ind()`, `ng_l2cap_lp_qos_req()`, `ng_l2cap_lp_receive()`, `ng_l2cap_process_command_timeout()`, `ng_l2cap_process_discon_timeout()`, and `ng_l2cap_process_lp_timeout()`.

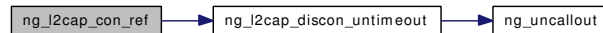
7.60.2.7 `void ng_l2cap_con_ref (ng_l2cap_con_p)`

Definition at line 150 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_CON_OUTGOING`, `ng_l2cap_discon_untimeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_con::refcnt`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_new_chan()`.

Here is the call graph for this function:



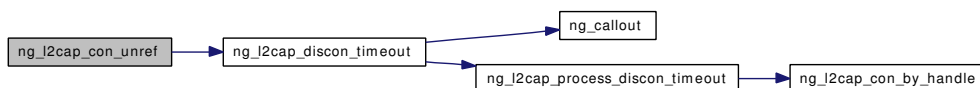
7.60.2.8 `void ng_l2cap_con_unref (ng_l2cap_con_p)`

Definition at line 171 of file `ng_l2cap_misc.c`.

References `ng_l2cap::discon_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_DYING`, `NG_L2CAP_CON_OPEN`, `NG_L2CAP_CON_OUTGOING`, `ng_l2cap_discon_timeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_con::refcnt`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_free_chan()`.

Here is the call graph for this function:



7.60.2.9 [ng_l2cap_flow_p ng_l2cap_default_flow \(void\)](#)

Definition at line 573 of file ng_l2cap_misc.c.

References NG_HCI_SERVICE_TYPE_BEST_EFFORT.

Referenced by ng_l2cap_l2ca_cfg_req(), and ng_l2cap_new_chan().

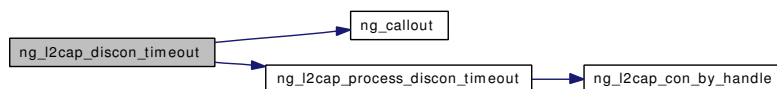
7.60.2.10 [int ng_l2cap_discon_timeout \(ng_l2cap_con_p\)](#)

Definition at line 202 of file ng_l2cap_misc.c.

References ng_l2cap_con::con_handle, ng_l2cap_con::con_timo, ng_l2cap::discon_timo, ng_l2cap_con::flags, ng_l2cap_con::l2cap, ng_callout(), NG_L2CAP_CON_AUTO_DISCON_TIMO, NG_L2CAP_CON_LP_TIMO, ng_l2cap_process_discon_timeout(), NG_NODE_NAME, ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_con_unref().

Here is the call graph for this function:



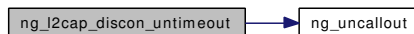
7.60.2.11 [int ng_l2cap_discon_untimeout \(ng_l2cap_con_p\)](#)

Definition at line 224 of file ng_l2cap_misc.c.

References ng_l2cap_con::con_timo, ng_l2cap_con::flags, ng_l2cap_con::l2cap, NG_L2CAP_CON_AUTO_DISCON_TIMO, NG_NODE_NAME, ng_uncallout(), ng_l2cap::node, and ng_l2cap_con::state.

Referenced by ng_l2cap_cleanup(), ng_l2cap_con_ref(), and ng_l2cap_lp_discon_ind().

Here is the call graph for this function:



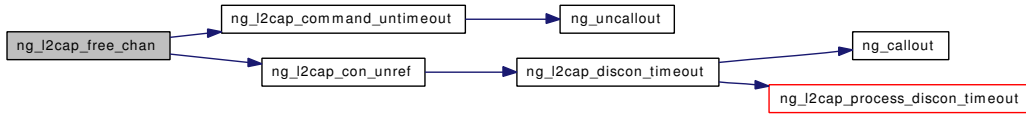
7.60.2.12 [void ng_l2cap_free_chan \(ng_l2cap_chan_p\)](#)

Definition at line 388 of file ng_l2cap_misc.c.

References ng_l2cap_cmd::ch, ng_l2cap_chan::con, M_NETGRAPH_L2CAP, NG_L2CAP_CMD_PENDING, ng_l2cap_command_untimeout(), ng_l2cap_con_unref(), ng_l2cap_free_cmd, and ng_l2cap_unlink_cmd.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_destroy_channels()`, `ng_l2cap_free_con()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, `ng_l2cap_process_con_req()`, `ng_l2cap_process_con_rsp()`, `ng_l2cap_process_discon_req()`, and `ng_l2cap_process_discon_rsp()`.

Here is the call graph for this function:



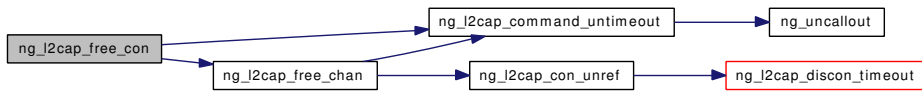
7.60.2.13 void ng_l2cap_free_con (ng_l2cap_con_p)

Definition at line 245 of file `ng_l2cap_misc.c`.

References `ng_l2cap_chan::con`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `M_NETGRAPH_L2CAP`, `NG_FREE_M`, `NG_L2CAP_CMD_PENDING`, `ng_l2cap_command_timeout()`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_L2CAP_CON_CLOSED`, `NG_L2CAP_CON_LP_TIMO`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `ng_l2cap_unlink_cmd`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_con::rx_pkt`, `ng_l2cap_con::state`, and `ng_l2cap_con::tx_pkt`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

Here is the call graph for this function:



7.60.2.14 u_int8_t ng_l2cap_get_ident (ng_l2cap_con_p)

Definition at line 622 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::ident`, `ng_l2cap_cmd_by_ident()`, `NG_L2CAP_FIRST_IDENT`, and `NG_L2CAP_NULL_IDENT`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, and `ng_l2cap_l2ca_ping_req()`.

Here is the call graph for this function:



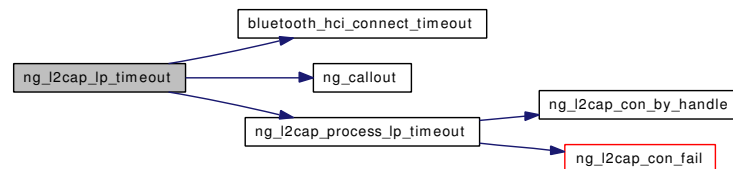
7.60.2.15 int ng_l2cap_lp_timeout (ng_l2cap_con_p)

Definition at line 471 of file `ng_l2cap_misc.c`.

References `bluetooth_hci_connect_timeout()`, `ng_l2cap_con::con_handle`, `ng_l2cap_con::con_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `ng_callout()`, `NG_L2CAP_CON_AUTO_DISCON_TIMO`, `NG_L2CAP_CON_LP_TIMO`, `ng_l2cap_process_lp_timeout()`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

Here is the call graph for this function:



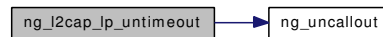
7.60.2.16 `int ng_l2cap_lp_untimeout (ng_l2cap_con_p)`

Definition at line 493 of file `ng_l2cap_misc.c`.

References `ng_l2cap_con::con_timo`, `ng_l2cap_con::flags`, `ng_l2cap_con::l2cap`, `NG_L2CAP_CON_LP_TIMO`, `NG_NODE_NAME`, `ng_uncallout()`, `ng_l2cap::node`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_cleanup()`, `ng_l2cap_lp_con_cfm()`, `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

Here is the call graph for this function:



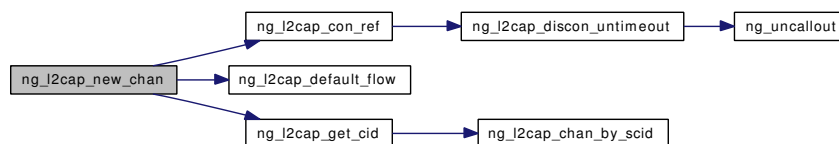
7.60.2.17 `ng_l2cap_chan_p ng_l2cap_new_chan (ng_l2cap_p, ng_l2cap_con_p, u_int16_t)`

Definition at line 328 of file `ng_l2cap_misc.c`.

References `M_NETGRAPH_L2CAP`, `NG_L2CAP_CLOSED`, `ng_l2cap_con_ref()`, `ng_l2cap_default_flow()`, `NG_L2CAP_FLUSH_TIMO_DEFAULT`, `ng_l2cap_get_cid()`, `NG_L2CAP_LINK_TIMO_DEFAULT`, `NG_L2CAP_MTU_DEFAULT`, `NG_L2CAP_NULL_CID`, and `ng_l2cap_con::state`.

Referenced by `ng_l2cap_l2ca_con_req()`, and `ng_l2cap_process_con_req()`.

Here is the call graph for this function:



7.60.2.18 `ng_l2cap_cmd_p ng_l2cap_new_cmd (ng_l2cap_con_p, ng_l2cap_chan_p, u_int8_t, u_int8_t, u_int32_t)`

Definition at line 419 of file `ng_l2cap_misc.c`.

References `ng_l2cap_chan::con`, `ng_l2cap_chan::ident`, `ng_l2cap_con::l2cap`, `M_NETGRAPH_L2CAP`, `ng_callout_init`, `NG_NODE_NAME`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_l2ca_cfg_req()`, `ng_l2cap_l2ca_cfg_rsp_req()`, `ng_l2cap_l2ca_con_req()`, `ng_l2cap_l2ca_con_rsp_req()`, `ng_l2cap_l2ca_discon_req()`, `ng_l2cap_l2ca_get_info_req()`, `ng_l2cap_l2ca_ping_req()`, `ng_l2cap_l2ca_write_req()`, `ng_l2cap_process_discon_req()`, `ng_l2cap_process_echo_req()`, `ng_l2cap_process_info_req()`, `send_l2cap_cfg_rsp()`, `send_l2cap_con_rej()`, and `send_l2cap_reject()`.

7.60.2.19 `ng_l2cap_con_p ng_l2cap_new_con (ng_l2cap_p, bdaddr_p)`

Definition at line 101 of file `ng_l2cap_misc.c`.

References `M_NETGRAPH_L2CAP`, `ng_callout_init`, `NG_L2CAP_CON_CLOSED`, and `NG_L2CAP_FIRST_IDENT`.

Referenced by `ng_l2cap_lp_con_ind()`, and `ng_l2cap_lp_con_req()`.

7.60.2.20 `struct mbuf* ng_l2cap_prepend (struct mbuf *, int)`

Definition at line 559 of file `ng_l2cap_misc.c`.

Referenced by `ng_l2cap_con_wakeup()`, `ng_l2cap_lp_send()`, and `ng_l2cap_process_echo_req()`.

7.60.2.21 `void ng_l2cap_send_hook_info (node_p, hook_p, void *, int)`

Definition at line 65 of file `ng_l2cap_misc.c`.

References `ng_l2cap::bdaddr`, `ng_msg::data`, `ng_l2cap::hci`, `NG_HCI_BDADDR_ANY`, `NG_HOOK_NAME`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_INFO`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_NODE_HOOK_INFO`, and `ng_l2cap::node`.

Referenced by `ng_l2cap_connect()`, and `ng_l2cap_lower_rcvmsg()`.

7.61 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_prse.h File Reference

This graph shows which files directly or indirectly include this file:



Variables

- static struct `ng_cmdlist ng_l2cap_cmdlist []`

7.61.1 Variable Documentation

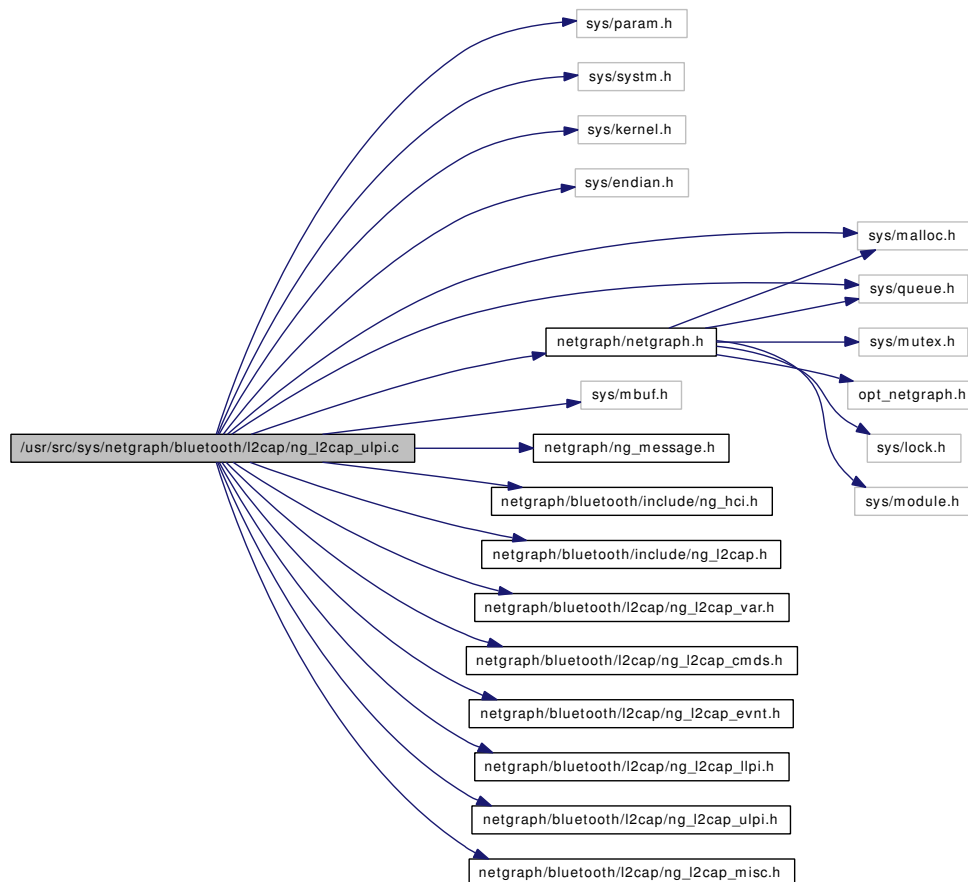
7.61.1.1 struct `ng_cmdlist ng_l2cap_cmdlist[]` [static]

Definition at line 47 of file `ng_l2cap_prse.h`.

7.62 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/endian.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_var.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_cmds.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_evnt.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_llpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h>
#include <netgraph/bluetooth/l2cap/ng_l2cap_misc.h>
```

Include dependency graph for ng_l2cap_ulpi.c:



Functions

- [int ng_l2cap_l2ca_con_req](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)
- [int ng_l2cap_l2ca_con_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result, u_int16_t status)
- [int ng_l2cap_l2ca_con_rsp_req](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)
- [int ng_l2cap_l2ca_con_rsp_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)
- [int ng_l2cap_l2ca_con_ind](#) (ng_l2cap_chan_p ch)
- [int ng_l2cap_l2ca_cfg_req](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)
- [int ng_l2cap_l2ca_cfg_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)
- [int ng_l2cap_l2ca_cfg_rsp_req](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)
- [int ng_l2cap_l2ca_cfg_rsp_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)
- [int ng_l2cap_l2ca_cfg_ind](#) (ng_l2cap_chan_p ch)
- [int ng_l2cap_l2ca_write_req](#) (ng_l2cap_p l2cap, struct [mbuf](#) *m)
- [int ng_l2cap_l2ca_write_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result, u_int16_t length)
- [int ng_l2cap_l2ca_receive](#) (ng_l2cap_con_p con)
- [int ng_l2cap_l2ca_clt_receive](#) (ng_l2cap_con_p con)
- [int ng_l2cap_l2ca_qos_ind](#) (ng_l2cap_chan_p ch)
- [int ng_l2cap_l2ca_discon_req](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)
- [int ng_l2cap_l2ca_discon_rsp](#) (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)
- [int ng_l2cap_l2ca_discon_ind](#) (ng_l2cap_chan_p ch)
- [int ng_l2cap_l2ca_grp_create](#) (ng_l2cap_p l2cap, struct [ng_mesg](#) *msg)

- `int ng_l2cap_l2ca_grp_close (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_grp_add_member_req (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_grp_add_member_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)`
- `int ng_l2cap_l2ca_grp_rem_member (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_grp_get_members (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_ping_req (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_ping_rsp (ng_l2cap_con_p con, u_int32_t token, u_int16_t result, struct mbuf *data)`
- `int ng_l2cap_l2ca_get_info_req (ng_l2cap_p l2cap, struct ng_mesg *msg)`
- `int ng_l2cap_l2ca_get_info_rsp (ng_l2cap_con_p con, u_int32_t token, u_int16_t result, struct mbuf *data)`
- `int ng_l2cap_l2ca_enable_clt (ng_l2cap_p l2cap, struct ng_mesg *msg)`

7.62.1 Function Documentation

7.62.1.1 `int ng_l2cap_l2ca_cfg_ind (ng_l2cap_chan_p ch)`

Definition at line 709 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::flush_timo`, `ng_l2cap_chan::iflow`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG_IND`, `ng_l2cap::node`, `ng_l2cap_chan::omtu`, `ng_l2cap_chan::psm`, and `ng_l2cap_chan::scid`.

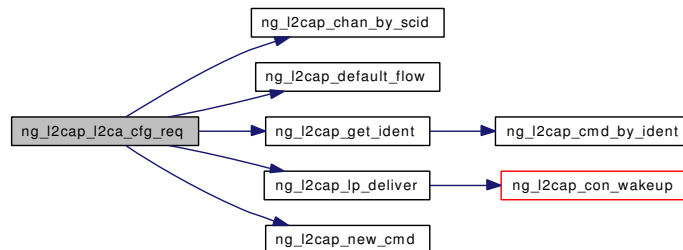
7.62.1.2 `int ng_l2cap_l2ca_cfg_req (ng_l2cap_p l2cap, struct ng_mesg *msg)`

Definition at line 390 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_build_cfg_options`, `_ng_l2cap_cfg_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_l2cap_chan::flush_timo`, `ng_mesg::header`, `ng_l2cap_chan::imtu`, `ng_l2cap_chan::link_timo`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `NG_L2CAP_CFG_REQ`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `ng_l2cap_default_flow()`, `NG_L2CAP_ERR`, `NG_L2CAP_FLUSH_TIMO_DEFAULT`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `NG_L2CAP_MTU_DEFAULT`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_OPEN`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::oflow`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.62.1.3 `int ng_l2cap_l2ca_cfg_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)`

Definition at line 497 of file ng_l2cap_ulpi.c.

References `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_l2cap_chan::flush_timo`, `ng_mesg::header`, `ng_l2cap_chan::imtu`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CFG_BOTH`, `NG_L2CAP_CFG_IN`, `NG_L2CAP_ERR`, `NG_L2CAP_OPEN`, `NG_L2CAP_SUCCESS`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG`, `ng_l2cap::node`, `ng_l2cap_chan::oflow`, `ng_l2cap_chan::psm`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, and `ng_l2cap_process_command_timeout()`.

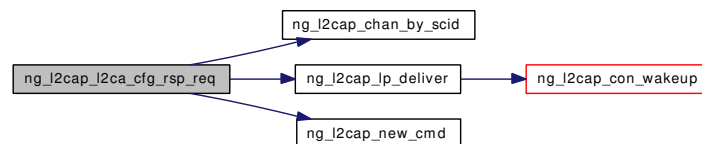
7.62.1.4 `int ng_l2cap_l2ca_cfg_rsp_req (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 560 of file ng_l2cap_ulpi.c.

References `_ng_l2cap_build_cfg_options`, `_ng_l2cap_cfg_rsp`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_mesg::header`, `ng_l2cap_chan::ident`, `ng_l2cap_chan::iflow`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `NG_L2CAP_CFG_BOTH`, `NG_L2CAP_CFG_OUT`, `NG_L2CAP_CFG_RSP`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_OPEN`, `NG_L2CAP_SUCCESS`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::omtu`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:

**7.62.1.5** `int ng_l2cap_l2ca_cfg_rsp_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)`

Definition at line 656 of file ng_l2cap_ulpi.c.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG_RSP`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_wakeup()`.

7.62.1.6 `int ng_l2cap_l2ca_clt_receive (ng_l2cap_con_p con)`

Definition at line 956 of file ng_l2cap_ulpi.c.

References `ng_l2cap::flags`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CLT_RFCOMM_DISABLED`, `NG_L2CAP_CLT_SDP_DISABLED`,

NG_L2CAP_CLT_TCP_DISABLED, NG_L2CAP_ERR, NG_L2CAP_M_PULLUP, NG_L2CAP_MTU_DEFAULT, NG_L2CAP_PSM_RFCOMM, NG_L2CAP_PSM_SDP, NG_L2CAP_PSM_TCP, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_l2cap::node, packed, and ng_l2cap_con::rx_pkt.

Referenced by ng_l2cap_receive().

7.62.1.7 int ng_l2cap_l2ca_con_ind (ng_l2cap_chan_p ch)

Definition at line 349 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_l2cap_chan::ident, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_CON_IND, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_con::remote, and ng_l2cap_chan::scid.

Referenced by ng_l2cap_process_con_req().

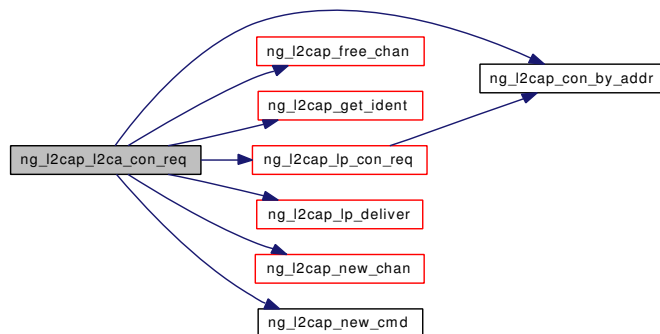
7.62.1.8 int ng_l2cap_l2ca_con_req (ng_l2cap_p l2cap, struct ng_mesg * msg)

Definition at line 63 of file ng_l2cap_ulpi.c.

References _ng_l2cap_con_req, ng_mesg::ng_msghdr::arglen, ng_l2cap::bdaddr, ng_l2cap_chan::con, ng_mesg::data, ng_mesg::header, NG_L2CAP_ALERT, ng_l2cap_con_by_addr(), NG_L2CAP_CON_REQ, NG_L2CAP_ERR, ng_l2cap_free_chan(), ng_l2cap_free_cmd, ng_l2cap_get_ident(), ng_l2cap_link_cmd, ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_new_chan(), ng_l2cap_new_cmd(), NG_L2CAP_NULL_IDENT, NG_L2CAP_W4_L2CAP_CON_RSP, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_chan::scid, ng_l2cap_chan::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_upper_rcvmsg().

Here is the call graph for this function:



7.62.1.9 int ng_l2cap_l2ca_con_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result, u_int16_t status)

Definition at line 151 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_mesg::ng_msghdr::flags, ng_mesg::header, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE,

NG_NODE_NAME, NG_SEND_MSG_HOOK, NGF_RESP, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_CON, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_chan::scid, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_con_rsp().

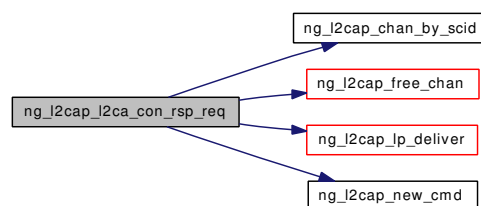
7.62.1.10 int ng_l2cap_l2ca_con_rsp_req (ng_l2cap_p l2cap, struct ng_mesg * msg)

Definition at line 201 of file ng_l2cap_ulpi.c.

References _ng_l2cap_con_rsp, ng_mesg::ng_msghdr::arglen, ng_l2cap_chan::cfg_state, ng_l2cap_chan::con, ng_mesg::data, ng_l2cap_chan::dcid, ng_mesg::header, ng_l2cap_chan::ident, NG_L2CAP_ALERT, ng_l2cap_chan_by_scid(), NG_L2CAP_CON_RSP, NG_L2CAP_CONFIG, NG_L2CAP_ERR, ng_l2cap_free_chan(), ng_l2cap_free_cmd, ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), ng_l2cap_new_cmd(), NG_L2CAP_PENDING, NG_L2CAP_SUCCESS, NG_L2CAP_W4_L2CA_CON_RSP, NG_L2CAP_WARN, NG_NODE_NAME, ng_l2cap::node, ng_l2cap_chan::scid, ng_l2cap_chan::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_upper_rcvmsg().

Here is the call graph for this function:



7.62.1.11 int ng_l2cap_l2ca_con_rsp_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)

Definition at line 309 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_mesg::ng_msghdr::flags, ng_mesg::header, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGF_RESP, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_CON_RSP, ng_l2cap::node, ng_l2cap_chan::psm, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_con_fail(), and ng_l2cap_con_wakeup().

7.62.1.12 int ng_l2cap_l2ca_discon_ind (ng_l2cap_chan_p ch)

Definition at line 1195 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_DISCON_IND, ng_l2cap::node, ng_l2cap_chan::psm, and ng_l2cap_chan::scid.

Referenced by ng_l2cap_con_fail(), and ng_l2cap_process_discon_req().

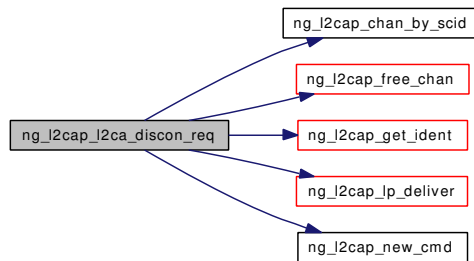
7.62.1.13 `int ng_l2cap_l2ca_discon_req (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1075 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_discon_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_ERR`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_OPEN`, `NG_L2CAP_W4_L2CAP_DISCON_RSP`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.62.1.14 `int ng_l2cap_l2ca_discon_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)`

Definition at line 1155 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_DISCON`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_discon_rsp()`.

7.62.1.15 `int ng_l2cap_l2ca_enable_clt (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1548 of file `ng_l2cap_ulpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::ctl`, `ng_mesg::data`, `ng_l2cap::flags`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_HOOK_IS_VALID`, `NG_L2CAP_ALERT`, `NG_L2CAP_CLT_RFCOMM_DISABLED`, `NG_L2CAP_CLT_SDP_DISABLED`, `NG_L2CAP_CLT_TCP_DISABLED`, `NG_L2CAP_ERR`, `NG_L2CAP_PSM_NOT_SUPPORTED`, `NG_L2CAP_PSM_RFCOMM`, `NG_L2CAP_PSM_SDP`, `NG_L2CAP_PSM_TCP`, `NG_L2CAP_SUCCESS`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_ENABLE_CLT`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

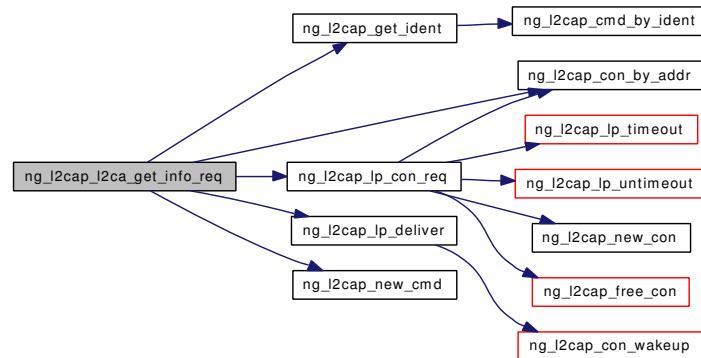
7.62.1.16 `int ng_l2cap_l2ca_get_info_req (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1427 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_info_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `NG_L2CAP_INFO_REQ`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.62.1.17 `int ng_l2cap_l2ca_get_info_rsp (ng_l2cap_con_p con, u_int32_t token, u_int16_t result, struct mbuf * data)`

Definition at line 1498 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap::ctl`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_WARN`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_GET_INFO`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_info_rsp()`.

7.62.1.18 `int ng_l2cap_l2ca_grp_add_member_req (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1254 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.62.1.19 `int ng_l2cap_l2ca_grp_add_member_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result)`

Definition at line 1265 of file `ng_l2cap_ulpi.c`.

7.62.1.20 `int ng_l2cap_l2ca_grp_close (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1243 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.62.1.21 `int ng_l2cap_l2ca_grp_create (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1232 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.62.1.22 `int ng_l2cap_l2ca_grp_get_members (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1288 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.62.1.23 `int ng_l2cap_l2ca_grp_rem_member (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1277 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

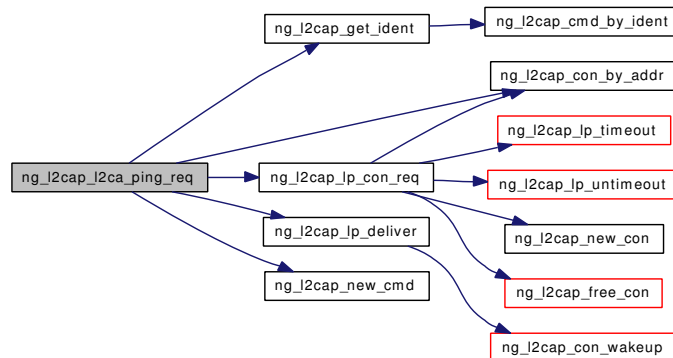
7.62.1.24 `int ng_l2cap_l2ca_ping_req (ng_l2cap_p l2cap, struct ng_mesg * msg)`

Definition at line 1298 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_echo_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_ECHO_REQ`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `NG_L2CAP_MAX_ECHO_SIZE`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_WARN`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.62.1.25 `int ng_l2cap_l2ca_ping_rsp (ng_l2cap_con_p con, u_int32_t token, u_int16_t result, struct mbuf * data)`

Definition at line 1377 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap::ctl`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_WARN`, `NG_MKMESSAGE`, `NG_`

NODE_NAME, NG_SEND_MSG_HOOK, NGF_RESP, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_PING, ng_l2cap::node, ng_l2cap_con::remote, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), and ng_l2cap_process_echo_rsp().

7.62.1.26 int ng_l2cap_l2ca_qos_ind (ng_l2cap_chan_p ch)

Definition at line 1039 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_QOS_IND, ng_l2cap::node, ng_l2cap_chan::psm, and ng_l2cap_con::remote.

7.62.1.27 int ng_l2cap_l2ca_receive (ng_l2cap_con_p con)

Definition at line 882 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::imtu, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_FREE_M, NG_HOOK_NOT_VALID, ng_l2cap_chan_by_scid(), NG_L2CAP_ERR, NG_L2CAP_M_PULLUP, NG_L2CAP_OPEN, NG_L2CAP_WARN, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_con::rx_pkt, ng_l2cap_chan::scid, and ng_l2cap_chan::state.

Referenced by ng_l2cap_receive().

Here is the call graph for this function:



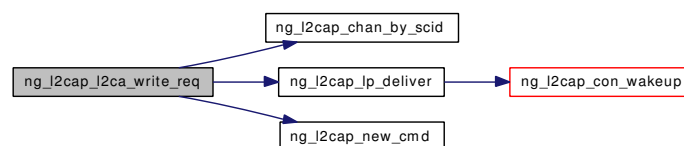
7.62.1.28 int ng_l2cap_l2ca_write_req (ng_l2cap_p l2cap, struct mbuf * m)

Definition at line 749 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, NG_FREE_M, ng_l2cap_chan_by_scid(), NG_L2CAP_ERR, NG_L2CAP_FIRST_CID, ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), NG_L2CAP_M_PULLUP, ng_l2cap_new_cmd(), NG_L2CAP_OPEN, NG_NODE_NAME, NGM_L2CAP_L2CA_WRITE, ng_l2cap::node, ng_l2cap_chan::scid, and ng_l2cap_chan::state.

Referenced by ng_l2cap_rcvdata().

Here is the call graph for this function:



7.62.1.29 `int ng_l2cap_l2ca_write_rsp (ng_l2cap_chan_p ch, u_int32_t token, u_int16_t result, u_int16_t length)`

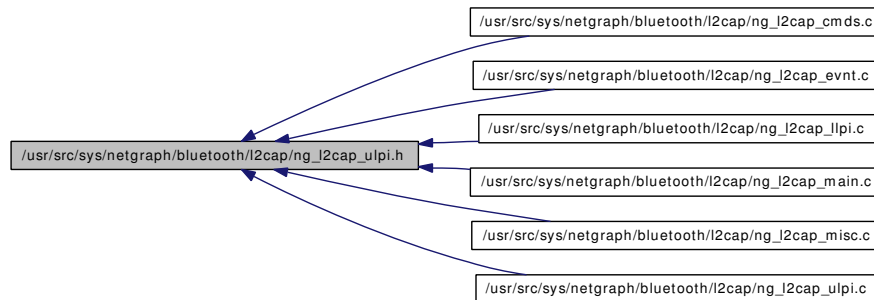
Definition at line 838 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_WRITE`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, `ng_l2cap_chan::scid`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_wakeup()`.

7.63 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int ng_l2cap_l2ca_con_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_con_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t, u_int16_t)`
- `int ng_l2cap_l2ca_con_rsp_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_con_rsp_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`
- `int ng_l2cap_l2ca_con_ind (ng_l2cap_chan_p)`
- `int ng_l2cap_l2ca_cfg_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_cfg_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`
- `int ng_l2cap_l2ca_cfg_rsp_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_cfg_rsp_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`
- `int ng_l2cap_l2ca_cfg_ind (ng_l2cap_chan_p)`
- `int ng_l2cap_l2ca_write_req (ng_l2cap_p, struct mbuf *)`
- `int ng_l2cap_l2ca_write_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t, u_int16_t)`
- `int ng_l2cap_l2ca_receive (ng_l2cap_con_p)`
- `int ng_l2cap_l2ca_clt_receive (ng_l2cap_con_p)`
- `int ng_l2cap_l2ca_qos_ind (ng_l2cap_chan_p)`
- `int ng_l2cap_l2ca_discon_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_discon_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`
- `int ng_l2cap_l2ca_discon_ind (ng_l2cap_chan_p)`
- `int ng_l2cap_l2ca_grp_create (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_grp_close (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_grp_add_member_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_grp_add_member_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`
- `int ng_l2cap_l2ca_grp_rem_member (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_grp_get_members (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_ping_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_ping_rsp (ng_l2cap_con_p, u_int32_t, u_int16_t, struct mbuf *)`
- `int ng_l2cap_l2ca_get_info_req (ng_l2cap_p, struct ng_mesg *)`
- `int ng_l2cap_l2ca_get_info_rsp (ng_l2cap_con_p, u_int32_t, u_int16_t, struct mbuf *)`
- `int ng_l2cap_l2ca_enable_clt (ng_l2cap_p, struct ng_mesg *)`

7.63.1 Function Documentation

7.63.1.1 `int ng_l2cap_l2ca_cfg_ind (ng_l2cap_chan_p)`

Definition at line 709 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::flush_timo`, `ng_l2cap_chan::iflow`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG_IND`, `ng_l2cap::node`, `ng_l2cap_chan::omtu`, `ng_l2cap_chan::psm`, and `ng_l2cap_chan::scid`.

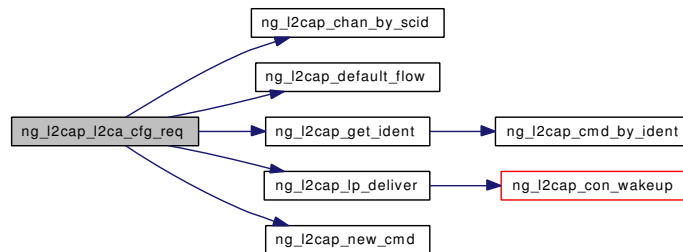
7.63.1.2 `int ng_l2cap_l2ca_cfg_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 390 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_build_cfg_options`, `_ng_l2cap_cfg_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_l2cap_chan::flush_timo`, `ng_mesg::header`, `ng_l2cap_chan::imtu`, `ng_l2cap_chan::link_timo`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `NG_L2CAP_CFG_REQ`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `ng_l2cap_default_flow()`, `NG_L2CAP_ERR`, `NG_L2CAP_FLUSH_TIMO_DEFAULT`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `NG_L2CAP_MTU_DEFAULT`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_OPEN`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::oflow`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.3 `int ng_l2cap_l2ca_cfg_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`

Definition at line 497 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_l2cap_chan::flush_timo`, `ng_mesg::header`, `ng_l2cap_chan::imtu`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CFG_BOTH`, `NG_L2CAP_CFG_IN`, `NG_L2CAP_ERR`, `NG_L2CAP_OPEN`, `NG_L2CAP_SUCCESS`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG`, `ng_l2cap::node`, `ng_l2cap_chan::oflow`, `ng_l2cap_chan::psm`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, and `ng_l2cap_process_command_timeout()`.

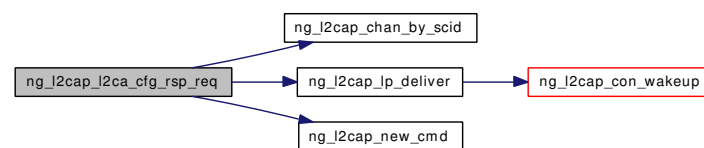
7.63.1.4 `int ng_l2cap_l2ca_cfg_rsp_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 560 of file ng_l2cap_ulpi.c.

References `_ng_l2cap_build_cfg_options`, `_ng_l2cap_cfg_rsp`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_mesg::header`, `ng_l2cap_chan::ident`, `ng_l2cap_chan::iflow`, `NG_FREE_M`, `NG_L2CAP_ALERT`, `NG_L2CAP_CFG_BOTH`, `NG_L2CAP_CFG_OUT`, `NG_L2CAP_CFG_RSP`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_OPEN`, `NG_L2CAP_SUCCESS`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::omtu`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:

**7.63.1.5** `int ng_l2cap_l2ca_cfg_rsp_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`

Definition at line 656 of file ng_l2cap_ulpi.c.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG_RSP`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_wakeup()`.

7.63.1.6 `int ng_l2cap_l2ca_clt_receive (ng_l2cap_con_p)`

Definition at line 956 of file ng_l2cap_ulpi.c.

References `ng_l2cap::flags`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_CLT_RFCOMM_DISABLED`, `NG_L2CAP_CLT_SDP_DISABLED`, `NG_L2CAP_CLT_TCP_DISABLED`, `NG_L2CAP_ERR`, `NG_L2CAP_M_PULLUP`, `NG_L2CAP_MTMU_DEFAULT`, `NG_L2CAP_PSM_RFCOMM`, `NG_L2CAP_PSM_SDP`, `NG_L2CAP_PSM_TCP`, `NG_NODE_NAME`, `NG_SEND_DATA_ONLY`, `ng_l2cap::node`, `packed`, and `ng_l2cap_con::rx_pkt`.

Referenced by `ng_l2cap_receive()`.

7.63.1.7 `int ng_l2cap_l2ca_con_ind (ng_l2cap_chan_p)`

Definition at line 349 of file ng_l2cap_ulpi.c.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::ident`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CON_IND`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, `ng_l2cap_con::remote`, and `ng_l2cap_chan::scid`.

Referenced by `ng_l2cap_process_con_req()`.

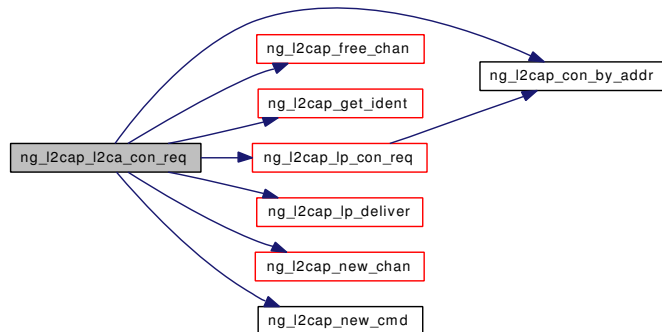
7.63.1.8 `int ng_l2cap_l2ca_con_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 63 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_con_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_CON_REQ`, `NG_L2CAP_ERR`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_chan()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_W4_L2CAP_CON_RSP`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.9 `int ng_l2cap_l2ca_con_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t, u_int16_t)`

Definition at line 151 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CON`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, `ng_l2cap_chan::scid`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_con_rsp()`.

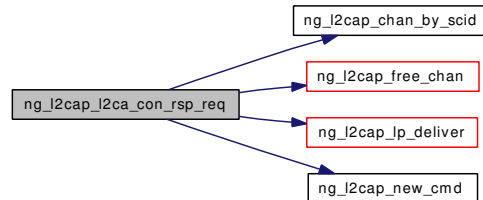
7.63.1.10 `int ng_l2cap_l2ca_con_rsp_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 201 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_con_rsp`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::cfg_state`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_mesg::header`, `ng_l2cap_chan::ident`, `NG_L2CAP_ALERT`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CON_RSP`, `NG_L2CAP_CONFIG`, `NG_L2CAP_ERR`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_PENDING`, `NG_L2CAP_SUCCESS`, `NG_L2CAP_W4_L2CA_CON_RSP`, `NG_L2CAP_WARN`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.11 `int ng_l2cap_l2ca_con_rsp_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`

Definition at line 309 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CON_RSP`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, and `ng_l2cap_con_wakeup()`.

7.63.1.12 `int ng_l2cap_l2ca_discon_ind (ng_l2cap_chan_p)`

Definition at line 1195 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_DISCON_IND`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_l2cap_chan::scid`.

Referenced by `ng_l2cap_con_fail()`, and `ng_l2cap_process_discon_req()`.

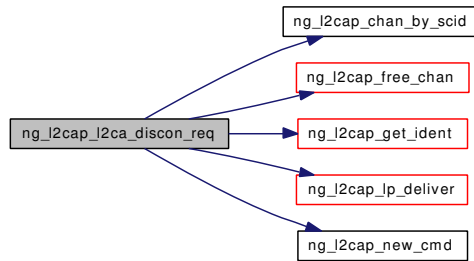
7.63.1.13 `int ng_l2cap_l2ca_discon_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1075 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_discon_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap_chan::dcid`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_chan_by_scid()`, `NG_L2CAP_CONFIG`, `NG_L2CAP_DISCON_REQ`, `NG_L2CAP_ERR`, `ng_l2cap_free_chan()`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_OPEN`, `NG_L2CAP_W4_L2CAP_DISCON_RSP`, `NG_NODE_NAME`, `ng_l2cap::node`, `ng_l2cap_chan::scid`, `ng_l2cap_chan::state`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.14 `int ng_l2cap_l2ca_discon_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`

Definition at line 1155 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_DISCON`, `ng_l2cap::node`, `ng_l2cap_chan::psm`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_discon_rsp()`.

7.63.1.15 `int ng_l2cap_l2ca_enable_clt (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1548 of file `ng_l2cap_ulpi.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::ctl`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_l2cap::flags`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_HOOK_IS_VALID`, `NG_L2CAP_ALERT`, `NG_L2CAP_CLT_RFCOMM_DISABLED`, `NG_L2CAP_CLT_SDP_DISABLED`, `NG_L2CAP_CLT_TCP_DISABLED`, `NG_L2CAP_ERR`, `NG_L2CAP_PSM_NOT_SUPPORTED`, `NG_L2CAP_PSM_RFCOMM`, `NG_L2CAP_PSM_SDP`, `NG_L2CAP_PSM_TCP`, `NG_L2CAP_SUCCESS`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_ENABLE_CLT`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

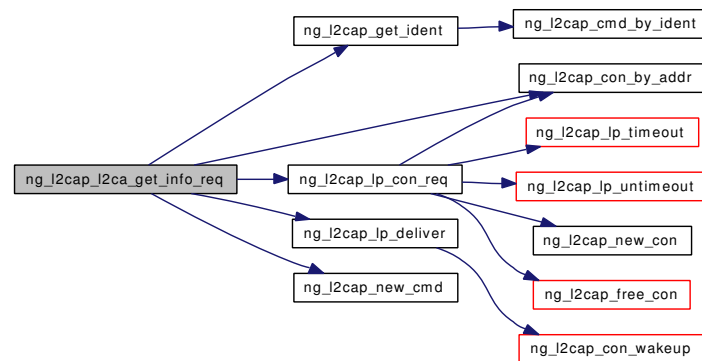
7.63.1.16 `int ng_l2cap_l2ca_get_info_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1427 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_info_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `NG_L2CAP_INFO_REQ`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.17 `int ng_l2cap_l2ca_get_info_rsp (ng_l2cap_con_p, u_int32_t, u_int16_t, struct mbuf *)`

Definition at line 1498 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap::ctl`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_WARN`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_GET_INFO`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_info_rsp()`.

7.63.1.18 `int ng_l2cap_l2ca_grp_add_member_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1254 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.63.1.19 `int ng_l2cap_l2ca_grp_add_member_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t)`

Definition at line 1265 of file `ng_l2cap_ulpi.c`.

7.63.1.20 `int ng_l2cap_l2ca_grp_close (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1243 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.63.1.21 `int ng_l2cap_l2ca_grp_create (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1232 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.63.1.22 `int ng_l2cap_l2ca_grp_get_members (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1288 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

7.63.1.23 `int ng_l2cap_l2ca_grp_rem_member (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1277 of file `ng_l2cap_ulpi.c`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

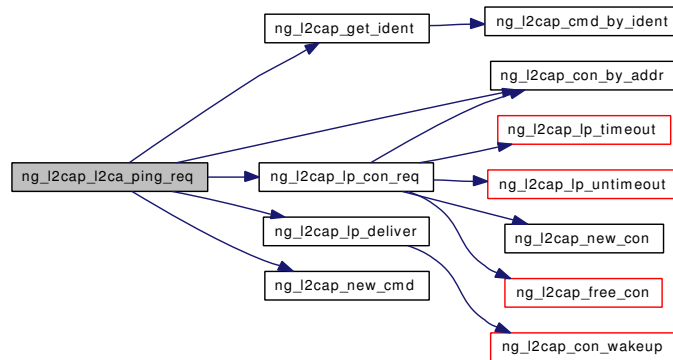
7.63.1.24 `int ng_l2cap_l2ca_ping_req (ng_l2cap_p, struct ng_mesg *)`

Definition at line 1298 of file `ng_l2cap_ulpi.c`.

References `_ng_l2cap_echo_req`, `ng_mesg::ng_msghdr::arglen`, `ng_l2cap::bdaddr`, `ng_mesg::data`, `ng_mesg::header`, `NG_L2CAP_ALERT`, `ng_l2cap_con_by_addr()`, `NG_L2CAP_ECHO_REQ`, `NG_L2CAP_ERR`, `ng_l2cap_free_cmd`, `ng_l2cap_get_ident()`, `ng_l2cap_link_cmd`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_deliver()`, `NG_L2CAP_MAX_ECHO_SIZE`, `ng_l2cap_new_cmd()`, `NG_L2CAP_NULL_IDENT`, `NG_L2CAP_WARN`, `NG_NODE_NAME`, `ng_l2cap::node`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_upper_rcvmsg()`.

Here is the call graph for this function:



7.63.1.25 `int ng_l2cap_l2ca_ping_rsp (ng_l2cap_con_p, u_int32_t, u_int16_t, struct mbuf *)`

Definition at line 1377 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap::ctl`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `ng_l2cap_con::l2cap`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_WARN`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`, `NGF_RESP`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_PING`, `ng_l2cap::node`, `ng_l2cap_con::remote`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_l2cap_con_fail()`, `ng_l2cap_con_wakeup()`, `ng_l2cap_process_cmd_rej()`, `ng_l2cap_process_command_timeout()`, and `ng_l2cap_process_echo_rsp()`.

7.63.1.26 `int ng_l2cap_l2ca_qos_ind (ng_l2cap_chan_p)`

Definition at line 1039 of file `ng_l2cap_ulpi.c`.

References `ng_l2cap_chan::con`, `ng_mesg::data`, `ng_l2cap::l2c`, `ng_l2cap_con::l2cap`, `NG_HOOK_NOT_VALID`, `NG_L2CAP_ERR`, `NG_MKMESSAGE`, `NG_NODE_NAME`, `NG_SEND_MSG_HOOK`,

NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_QOS_IND, ng_l2cap::node, ng_l2cap_chan::psm, and ng_l2cap_con::remote.

7.63.1.27 int ng_l2cap_l2ca_receive (ng_l2cap_con_p)

Definition at line 882 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::imtu, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_FREE_M, NG_HOOK_NOT_VALID, ng_l2cap_chan_by_scid(), NG_L2CAP_ERR, NG_L2CAP_M_PULLUP, NG_L2CAP_OPEN, NG_L2CAP_WARN, NG_NODE_NAME, NG_SEND_DATA_ONLY, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_con::rx_pkt, ng_l2cap_chan::scid, and ng_l2cap_chan::state.

Referenced by ng_l2cap_receive().

Here is the call graph for this function:



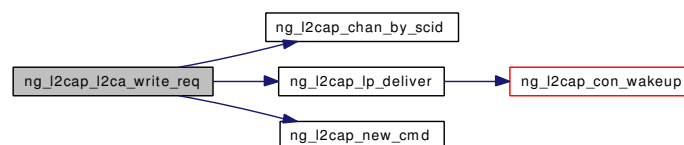
7.63.1.28 int ng_l2cap_l2ca_write_req (ng_l2cap_p, struct mbuf *)

Definition at line 749 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, NG_FREE_M, ng_l2cap_chan_by_scid(), NG_L2CAP_ERR, NG_L2CAP_FIRST_CID, ng_l2cap_link_cmd, ng_l2cap_lp_deliver(), NG_L2CAP_M_PULLUP, ng_l2cap_new_cmd(), NG_L2CAP_OPEN, NG_NODE_NAME, NGM_L2CAP_L2CA_WRITE, ng_l2cap::node, ng_l2cap_chan::scid, and ng_l2cap_chan::state.

Referenced by ng_l2cap_rcvdata().

Here is the call graph for this function:



7.63.1.29 int ng_l2cap_l2ca_write_rsp (ng_l2cap_chan_p, u_int32_t, u_int16_t, u_int16_t)

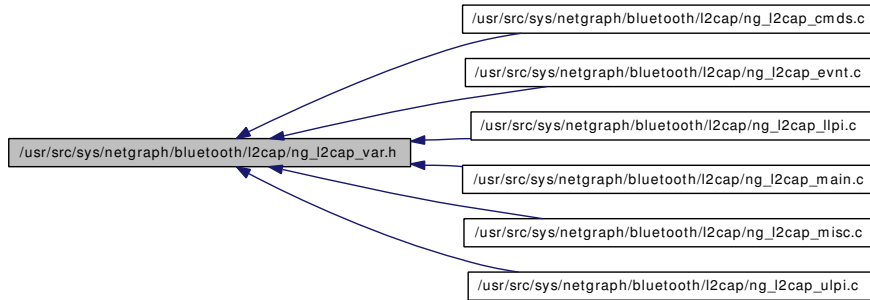
Definition at line 838 of file ng_l2cap_ulpi.c.

References ng_l2cap_chan::con, ng_mesg::data, ng_mesg::ng_msghdr::flags, ng_mesg::header, ng_l2cap::l2c, ng_l2cap_con::l2cap, NG_HOOK_NOT_VALID, NG_L2CAP_ERR, NG_MKMESSAGE, NG_NODE_NAME, NG_SEND_MSG_HOOK, NGF_RESP, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_WRITE, ng_l2cap::node, ng_l2cap_chan::psm, ng_l2cap_chan::scid, and ng_mesg::ng_msghdr::token.

Referenced by ng_l2cap_con_wakeup().

7.64 /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_l2cap](#)
- struct [ng_l2cap_con](#)
- struct [ng_l2cap_chan](#)
- struct [ng_l2cap_cmd](#)

Defines

- #define [M_NETGRAPH_L2CAP](#) M_NETGRAPH
- #define [NG_L2CAP_ALERT](#) if (l2cap → debug >= NG_L2CAP_ALERT_LEVEL) printf
- #define [NG_L2CAP_ERR](#) if (l2cap → debug >= NG_L2CAP_ERR_LEVEL) printf
- #define [NG_L2CAP_WARN](#) if (l2cap → debug >= NG_L2CAP_WARN_LEVEL) printf
- #define [NG_L2CAP_INFO](#) if (l2cap → debug >= NG_L2CAP_INFO_LEVEL) printf
- #define [NG_L2CAP_M_PULLUP](#)(m, s)
- #define [NG_L2CAP_NULL_IDENT](#) 0x00
- #define [NG_L2CAP_FIRST_IDENT](#) 0x01
- #define [NG_L2CAP_LAST_IDENT](#) 0xff
- #define [NG_L2CAP_CFG_IN](#) (1 << 0)
- #define [NG_L2CAP_CFG_OUT](#) (1 << 1)
- #define [NG_L2CAP_CFG_BOTH](#) (NG_L2CAP_CFG_IN|NG_L2CAP_CFG_OUT)
- #define [NG_L2CAP_CMD_PENDING](#) (1 << 0)

Typedefs

- typedef [ng_l2cap](#) [ng_l2cap_t](#)
- typedef [ng_l2cap_t](#) * [ng_l2cap_p](#)
- typedef [ng_l2cap_con](#) [ng_l2cap_con_t](#)
- typedef [ng_l2cap_con_t](#) * [ng_l2cap_con_p](#)
- typedef [ng_l2cap_chan](#) [ng_l2cap_chan_t](#)
- typedef [ng_l2cap_chan_t](#) * [ng_l2cap_chan_p](#)
- typedef [ng_l2cap_cmd](#) [ng_l2cap_cmd_t](#)
- typedef [ng_l2cap_cmd_t](#) * [ng_l2cap_cmd_p](#)

7.64.1 Define Documentation

7.64.1.1 #define M_NETGRAPH_L2CAP M_NETGRAPH

Definition at line 41 of file ng_l2cap_var.h.

7.64.1.2 #define NG_L2CAP_ALERT if (l2cap → debug >= NG_L2CAP_ALERT_LEVEL) printf

Definition at line 45 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_ping_req(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_cfm(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_receive(), ng_l2cap_lp_send(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), ng_l2cap_process_discon_timeout(), ng_l2cap_process_echo_req(), and ng_l2cap_process_lp_timeout().

7.64.1.3 #define NG_L2CAP_CFG_BOTH (NG_L2CAP_CFG_IN|NG_L2CAP_CFG_OUT)

Definition at line 143 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_cfg_rsp(), and ng_l2cap_l2ca_cfg_rsp_req().

7.64.1.4 #define NG_L2CAP_CFG_IN (1 << 0)

Definition at line 141 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_cfg_rsp().

7.64.1.5 #define NG_L2CAP_CFG_OUT (1 << 1)

Definition at line 142 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_cfg_rsp_req().

7.64.1.6 #define NG_L2CAP_CMD_PENDING (1 << 0)

Definition at line 173 of file ng_l2cap_var.h.

Referenced by ng_l2cap_cmd_by_ident(), ng_l2cap_command_timeout(), ng_l2cap_command_untimeout(), ng_l2cap_con_fail(), ng_l2cap_con_wakeup(), ng_l2cap_free_chan(), ng_l2cap_free_con(), and ng_l2cap_process_command_timeout().

7.64.1.7 #define NG_L2CAP_ERR if (l2cap → debug >= NG_L2CAP_ERR_LEVEL) printf

Definition at line 46 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_cfg_ind(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_cfg_rsp_rsp(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_con_ind(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_con_rsp_rsp(), ng_l2cap_l2ca_discon_ind(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_

discon_rsp(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_req(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), ng_l2cap_lp_receive(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_discon_timeout(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_rsp(), ng_l2cap_process_signal_cmd(), and ng_l2cap_receive().

7.64.1.8 #define NG_L2CAP_FIRST_IDENT 0x01

Definition at line 68 of file ng_l2cap_var.h.

Referenced by ng_l2cap_get_ident(), and ng_l2cap_new_con().

7.64.1.9 #define NG_L2CAP_INFO if (l2cap → debug >= NG_L2CAP_INFO_LEVEL) printf

Definition at line 48 of file ng_l2cap_var.h.

Referenced by ng_l2cap_con_fail(), ng_l2cap_lower_rcvmsg(), ng_l2cap_lp_deliver(), ng_l2cap_lp_receive(), ng_l2cap_lp_send(), ng_l2cap_process_con_rsp(), and ng_l2cap_send_hook_info().

7.64.1.10 #define NG_L2CAP_LAST_IDENT 0xff

Definition at line 69 of file ng_l2cap_var.h.

7.64.1.11 #define NG_L2CAP_M_PULLUP(m, s)

Value:

```
do { \
    if ((m)->m_len < (s)) \
        (m) = m_pullup((m), (s)); \
    if ((m) == NULL) \
        NG_L2CAP_ALERT("%s: %s - m_pullup(%zd) failed\n", \
            __func__, NG_NODE_NAME(l2cap->node), (s)); \
} while (0)
```

Definition at line 51 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_req(), ng_l2cap_lp_receive(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), ng_l2cap_process_signal_cmd(), and ng_l2cap_receive().

7.64.1.12 #define NG_L2CAP_NULL_IDENT 0x00

Definition at line 67 of file ng_l2cap_var.h.

Referenced by ng_l2cap_get_ident(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_get_info_req(), and ng_l2cap_l2ca_ping_req().

7.64.1.13 #define NG_L2CAP_WARN if (l2cap → debug >= NG_L2CAP_WARN_LEVEL) printf

Definition at line 47 of file ng_l2cap_var.h.

Referenced by ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_receive(), ng_l2cap_lower_rcvmsg(), and ng_l2cap_process_info_rsp().

7.64.2 Typedef Documentation**7.64.2.1 typedef ng_l2cap_chan_t* ng_l2cap_chan_p**

Definition at line 162 of file ng_l2cap_var.h.

7.64.2.2 typedef struct ng_l2cap_chan ng_l2cap_chan_t**7.64.2.3 typedef ng_l2cap_cmd_t* ng_l2cap_cmd_p**

Definition at line 185 of file ng_l2cap_var.h.

7.64.2.4 typedef struct ng_l2cap_cmd ng_l2cap_cmd_t**7.64.2.5 typedef ng_l2cap_con_t* ng_l2cap_con_p**

Definition at line 129 of file ng_l2cap_var.h.

7.64.2.6 typedef struct ng_l2cap_con ng_l2cap_con_t**7.64.2.7 typedef ng_l2cap_t* ng_l2cap_p**

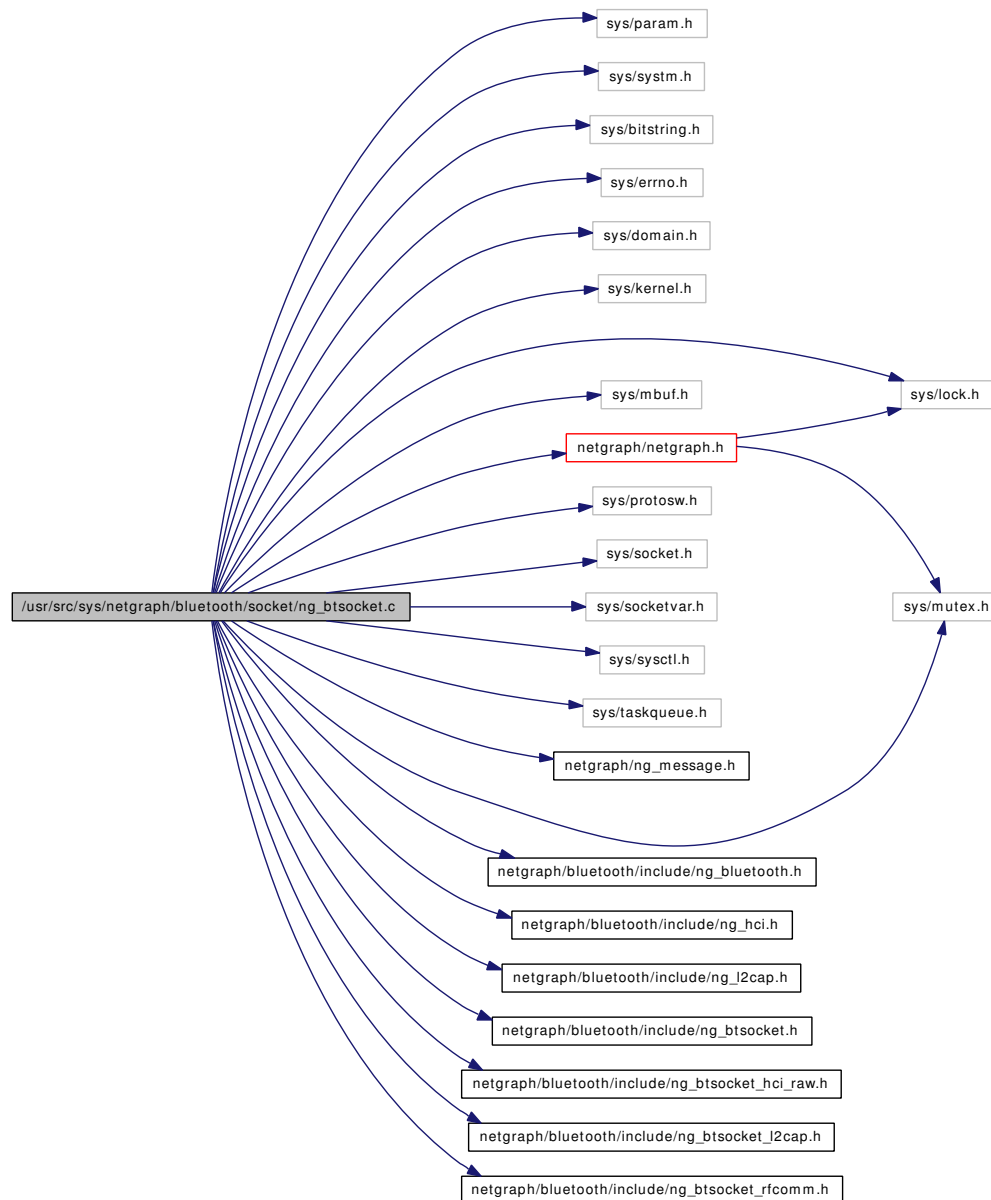
Definition at line 98 of file ng_l2cap_var.h.

7.64.2.8 typedef struct ng_l2cap ng_l2cap_t

7.65 /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bitstring.h>
#include <sys/errno.h>
#include <sys/domain.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/taskqueue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket.h>
#include <netgraph/bluetooth/include/ng_btsocket_hci_raw.h>
#include <netgraph/bluetooth/include/ng_btsocket_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket_rfcomm.h>
```

Include dependency graph for ng_btsocket.c:



Defines

- #define `ng_btsocket_protosw_size` (`sizeof(ng_btsocket_protosw)/sizeof(ng_btsocket_protosw[0])`)
- #define `ng_btsocket_protosw_end` `&ng_btsocket_protosw[ng_btsocket_protosw_size]`

Functions

- static int `ng_btsocket_modevent` (`module_t`, `int`, `void *`)
- `SYSCTL_NODE` (`_net_bluetooth_hci`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, `0`, "Bluetooth HCI sockets family")
- `SYSCTL_NODE` (`_net_bluetooth_l2cap`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, `0`, "Bluetooth L2CAP sockets family")

- **SYSCTL_NODE** (`_net_bluetooth_rfcomm`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, 0, "Bluetooth RFCOMM sockets family")
- **DECLARE_MODULE** (`ng_btsocket`, `ng_btsocket_mod`, `SI_SUB_PROTO_DOMAIN`, `SI_ORDER_ANY`)
- **MODULE_VERSION** (`ng_btsocket`, `NG_BLUETOOTH_VERSION`)
- **MODULE_DEPEND** (`ng_btsocket`, `ng_bluetooth`, `NG_BLUETOOTH_VERSION`, `NG_BLUETOOTH_VERSION`, `NG_BLUETOOTH_VERSION`)
- **MODULE_DEPEND** (`ng_btsocket`, `netgraph`, `NG_ABI_VERSION`, `NG_ABI_VERSION`, `NG_ABI_VERSION`)

Variables

- domain `ng_btsocket_domain`
- static struct `pr_usrreqs` `ng_btsocket_hci_raw_usrreqs`
- static struct `pr_usrreqs` `ng_btsocket_l2cap_raw_usrreqs`
- static struct `pr_usrreqs` `ng_btsocket_l2cap_usrreqs`
- static struct `pr_usrreqs` `ng_btsocket_rfcomm_usrreqs`
- static struct `protosw` `ng_btsocket_protosw` []
- domain `ng_btsocket_domain`
- static `moduledata_t` `ng_btsocket_mod`

7.65.1 Define Documentation

7.65.1.1 #define `ng_btsocket_protosw_end` &`ng_btsocket_protosw`[`ng_btsocket_protosw_size`]

Definition at line 184 of file `ng_btsocket.c`.

7.65.1.2 #define `ng_btsocket_protosw_size` (`sizeof(ng_btsocket_protosw)/sizeof(ng_btsocket_protosw[0])`)

Definition at line 182 of file `ng_btsocket.c`.

7.65.2 Function Documentation

7.65.2.1 **DECLARE_MODULE** (`ng_btsocket`, `ng_btsocket_mod`, `SI_SUB_PROTO_DOMAIN`, `SI_ORDER_ANY`)

7.65.2.2 **MODULE_DEPEND** (`ng_btsocket`, `netgraph`, `NG_ABI_VERSION`, `NG_ABI_VERSION`, `NG_ABI_VERSION`)

7.65.2.3 **MODULE_DEPEND** (`ng_btsocket`, `ng_bluetooth`, `NG_BLUETOOTH_VERSION`, `NG_BLUETOOTH_VERSION`, `NG_BLUETOOTH_VERSION`)

7.65.2.4 **MODULE_VERSION** (`ng_btsocket`, `NG_BLUETOOTH_VERSION`)

7.65.2.5 **static int** `ng_btsocket_modevent` (`module_t`, `int`, `void *`) [`static`]

Definition at line 233 of file `ng_btsocket.c`.

References `ng_btsocket_domain`.

7.65.2.6 SYSCTL_NODE (`_net_bluetooth_rfcomm`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, `0`, "Bluetooth RFCOMM sockets family")

7.65.2.7 SYSCTL_NODE (`_net_bluetooth_l2cap`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, `0`, "Bluetooth L2CAP sockets family")

7.65.2.8 SYSCTL_NODE (`_net_bluetooth_hci`, `OID_AUTO`, `sockets`, `CTLFLAG_RW`, `0`, "Bluetooth HCI sockets family")

7.65.3 Variable Documentation

7.65.3.1 struct domain `ng_btsocket_domain`

Initial value:

```
{
    .dom_family =          AF_BLUETOOTH,
    .dom_name =           "bluetooth",
    .dom_protosw =        ng_btsocket_protosw,
    .dom_protoswNPROTOSW = ng_btsocket_protosw_end
}
```

Definition at line 191 of file `ng_btsocket.c`.

Referenced by `ng_btsocket_modevent()`.

7.65.3.2 struct domain `ng_btsocket_domain`

Definition at line 191 of file `ng_btsocket.c`.

Referenced by `ng_btsocket_modevent()`.

7.65.3.3 struct pr_usrreqs `ng_btsocket_hci_raw_usrreqs` [static]

Initial value:

```
{
    .pru_abort =          ng_btsocket_hci_raw_abort,
    .pru_attach =        ng_btsocket_hci_raw_attach,
    .pru_bind =          ng_btsocket_hci_raw_bind,
    .pru_connect =       ng_btsocket_hci_raw_connect,
    .pru_control =       ng_btsocket_hci_raw_control,
    .pru_detach =        ng_btsocket_hci_raw_detach,
    .pru_disconnect =   ng_btsocket_hci_raw_disconnect,
    .pru_peeraddr =     ng_btsocket_hci_raw_peeraddr,
    .pru_send =          ng_btsocket_hci_raw_send,
    .pru_shutdown =     NULL,
    .pru_sockaddr =     ng_btsocket_hci_raw_sockaddr,
    .pru_close =         ng_btsocket_hci_raw_close,
}
```

Definition at line 65 of file `ng_btsocket.c`.

7.65.3.4 struct pr_usrreqs `ng_btsocket_l2cap_raw_usrreqs` [static]

Initial value:

```

{
    .pru_abort =          ng_btsocket_l2cap_raw_abort,
    .pru_attach =        ng_btsocket_l2cap_raw_attach,
    .pru_bind =          ng_btsocket_l2cap_raw_bind,
    .pru_connect =       ng_btsocket_l2cap_raw_connect,
    .pru_control =       ng_btsocket_l2cap_raw_control,
    .pru_detach =        ng_btsocket_l2cap_raw_detach,
    .pru_disconnect =    ng_btsocket_l2cap_raw_disconnect,
    .pru_peeraddr =      ng_btsocket_l2cap_raw_peeraddr,
    .pru_send =          ng_btsocket_l2cap_raw_send,
    .pru_shutdown =      NULL,
    .pru_sockaddr =      ng_btsocket_l2cap_raw_sockaddr,
    .pru_close =         ng_btsocket_l2cap_raw_close,
}

```

Definition at line 84 of file ng_btsocket.c.

7.65.3.5 struct pr_usrreqs ng_btsocket_l2cap_usrreqs [static]

Initial value:

```

{
    .pru_abort =          ng_btsocket_l2cap_abort,
    .pru_accept =         ng_btsocket_l2cap_accept,
    .pru_attach =         ng_btsocket_l2cap_attach,
    .pru_bind =           ng_btsocket_l2cap_bind,
    .pru_connect =        ng_btsocket_l2cap_connect,
    .pru_control =        ng_btsocket_l2cap_control,
    .pru_detach =         ng_btsocket_l2cap_detach,
    .pru_disconnect =     ng_btsocket_l2cap_disconnect,
    .pru_listen =         ng_btsocket_l2cap_listen,
    .pru_peeraddr =       ng_btsocket_l2cap_peeraddr,
    .pru_send =           ng_btsocket_l2cap_send,
    .pru_shutdown =       NULL,
    .pru_sockaddr =       ng_btsocket_l2cap_sockaddr,
    .pru_close =          ng_btsocket_l2cap_close,
}

```

Definition at line 103 of file ng_btsocket.c.

7.65.3.6 moduledata_t ng_btsocket_mod [static]

Initial value:

```

{
    "ng_btsocket",
    ng_btsocket_modevent,
    NULL
}

```

Definition at line 213 of file ng_btsocket.c.

7.65.3.7 struct protosw ng_btsocket_protosw[] [static]

Definition at line 145 of file ng_btsocket.c.

7.65.3.8 struct pru_usrreqs [ng_btsocket_rfcomm_usrreqs](#) [static]**Initial value:**

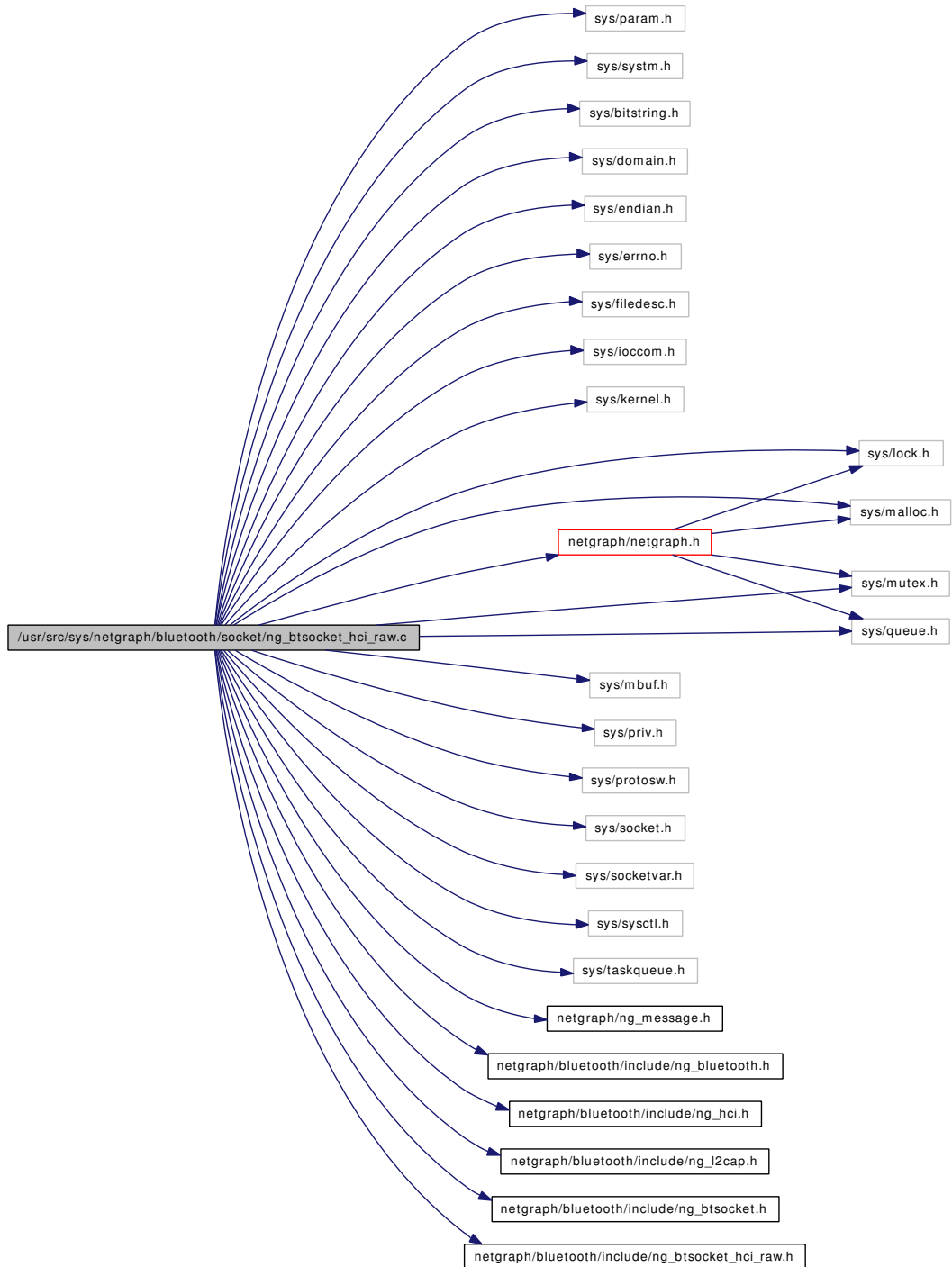
```
{
    .pru_abort =          ng_btsocket_rfcomm_abort,
    .pru_accept =        ng_btsocket_rfcomm_accept,
    .pru_attach =        ng_btsocket_rfcomm_attach,
    .pru_bind =          ng_btsocket_rfcomm_bind,
    .pru_connect =       ng_btsocket_rfcomm_connect,
    .pru_control =       ng_btsocket_rfcomm_control,
    .pru_detach =        ng_btsocket_rfcomm_detach,
    .pru_disconnect =    ng_btsocket_rfcomm_disconnect,
    .pru_listen =        ng_btsocket_rfcomm_listen,
    .pru_peeraddr =      ng_btsocket_rfcomm_peeraddr,
    .pru_send =          ng_btsocket_rfcomm_send,
    .pru_shutdown =      NULL,
    .pru_sockaddr =      ng_btsocket_rfcomm_sockaddr,
    .pru_close =         ng_btsocket_rfcomm_close,
}
```

Definition at line 124 of file `ng_btsocket.c`.

7.66 /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_hci_raw.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bitstring.h>
#include <sys/domain.h>
#include <sys/endian.h>
#include <sys/errno.h>
#include <sys/filedesc.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/taskqueue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket.h>
#include <netgraph/bluetooth/include/ng_btsocket_hci_raw.h>
```

Include dependency graph for ng_btsocket_hci_raw.c:



Data Structures

- struct [ng_btsocket_hci_raw_sec_filter](#)

Defines

- #define `M_NETGRAPH_BT SOCKET_HCI_RAW` `M_NETGRAPH`
- #define `ng_btsocket_hci_raw_wakeup_input_task()` `taskqueue_enqueue(taskqueue_swi, &ng_btsocket_hci_raw_task)`
- #define `NG_BT SOCKET_HCI_RAW_INFO`
- #define `NG_BT SOCKET_HCI_RAW_WARN`
- #define `NG_BT SOCKET_HCI_RAW_ERR`
- #define `NG_BT SOCKET_HCI_RAW_ALERT`

Functions

- static void `ng_btsocket_hci_raw_input` (void *, int)
- static void `ng_btsocket_hci_raw_output` (node_p, hook_p, void *, int)
- static void `ng_btsocket_hci_raw_savctl` (ng_btsocket_hci_raw_pcb_p, struct mbuf **, struct mbuf *)
- static int `ng_btsocket_hci_raw_filter` (ng_btsocket_hci_raw_pcb_p, struct mbuf *, int)
- static `LIST_HEAD` (ng_btsocket_hci_raw_pcb)
- static int `ng_btsocket_hci_raw_node_shutdown` (node_p node)
- static int `ng_btsocket_hci_raw_node_newhook` (node_p node, hook_p hook, char const *name)
- static int `ng_btsocket_hci_raw_node_connect` (hook_p hook)
- static int `ng_btsocket_hci_raw_node_disconnect` (hook_p hook)
- static int `ng_btsocket_hci_raw_node_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_btsocket_hci_raw_node_rcvdata` (hook_p hook, item_p item)
- static void `ng_btsocket_hci_raw_get_token` (u_int32_t *token)
- static int `ng_btsocket_hci_raw_send_ngmsg` (char *path, int cmd, void *arg, int arglen)
- static int `ng_btsocket_hci_raw_send_sync_ngmsg` (ng_btsocket_hci_raw_pcb_p pcb, char *path, int cmd, void *rsp, int rsplen)
- static void `ng_btsocket_hci_raw_data_input` (struct mbuf *nam)
- static void `ng_btsocket_hci_raw_msg_input` (struct ng_mesg *msg)
- void `ng_btsocket_hci_raw_init` (void)
- void `ng_btsocket_hci_raw_abort` (struct socket *so)
- void `ng_btsocket_hci_raw_close` (struct socket *so)
- int `ng_btsocket_hci_raw_attach` (struct socket *so, int proto, struct thread *td)
- int `ng_btsocket_hci_raw_bind` (struct socket *so, struct sockaddr *nam, struct thread *td)
- int `ng_btsocket_hci_raw_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- int `ng_btsocket_hci_raw_control` (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- int `ng_btsocket_hci_raw_ctloutput` (struct socket *so, struct sockopt *sopt)
- void `ng_btsocket_hci_raw_detach` (struct socket *so)
- int `ng_btsocket_hci_raw_disconnect` (struct socket *so)
- int `ng_btsocket_hci_raw_peeraddr` (struct socket *so, struct sockaddr **nam)
- int `ng_btsocket_hci_raw_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *sa, struct mbuf *control, struct thread *td)
- int `ng_btsocket_hci_raw_sockaddr` (struct socket *so, struct sockaddr **nam)

Variables

- static `ng_constructor_t` `ng_btsocket_hci_raw_node_constructor`
- static `ng_rcvmsg_t` `ng_btsocket_hci_raw_node_rcvmsg`
- static `ng_shutdown_t` `ng_btsocket_hci_raw_node_shutdown`
- static `ng_newhook_t` `ng_btsocket_hci_raw_node_newhook`
- static `ng_connect_t` `ng_btsocket_hci_raw_node_connect`
- static `ng_rcvdata_t` `ng_btsocket_hci_raw_node_rcvdata`
- static `ng_disconnect_t` `ng_btsocket_hci_raw_node_disconnect`
- static struct `ng_type` `typestruct`
- int `ifqmaxlen`
- static `u_int32_t` `ng_btsocket_hci_raw_debug_level`
- static `u_int32_t` `ng_btsocket_hci_raw_ioctl_timeout`
- static `node_p` `ng_btsocket_hci_raw_node`
- static struct `ng_bt_itemq` `ng_btsocket_hci_raw_queue`
- static struct `mtx` `ng_btsocket_hci_raw_queue_mtx`
- static struct `task` `ng_btsocket_hci_raw_task`

7.66.1 Define Documentation

7.66.1.1 #define M_NETGRAPH_BT_SOCKET_HCI_RAW M_NETGRAPH

Definition at line 67 of file `ng_btsocket_hci_raw.c`.

Referenced by `ng_btsocket_hci_raw_attach()`, `ng_btsocket_hci_raw_detach()`, and `ng_btsocket_hci_raw_init()`.

7.66.1.2 #define NG_BT_SOCKET_HCI_RAW_ALERT

Value:

```
if (ng_btsocket_hci_raw_debug_level >= NG_BT_SOCKET_ALERT_LEVEL) \
    printf
```

Referenced by `ng_btsocket_hci_raw_init()`, and `ng_btsocket_hci_raw_node_shutdown()`.

7.66.1.3 #define NG_BT_SOCKET_HCI_RAW_ERR

Value:

```
if (ng_btsocket_hci_raw_debug_level >= NG_BT_SOCKET_ERR_LEVEL) \
    printf
```

Referenced by `ng_btsocket_hci_raw_node_rcvdata()`, and `ng_btsocket_hci_raw_node_rcvmsg()`.

7.66.1.4 #define NG_BT_SOCKET_HCI_RAW_INFO

Value:

```
if (ng_btsocket_hci_raw_debug_level >= NG_BT_SOCKET_INFO_LEVEL) \
    printf
```

Referenced by `ng_btsocket_hci_raw_data_input()`.

7.66.1.5 `#define ng_btsocket_hci_raw_wakeup_input_task() taskqueue_enqueue(taskqueue_swi, &ng_btsocket_hci_raw_task)`

Definition at line 87 of file `ng_btsocket_hci_raw.c`.

Referenced by `ng_btsocket_hci_raw_node_rcvdata()`, and `ng_btsocket_hci_raw_node_rcvmsg()`.

7.66.1.6 `#define NG_BTSOCKET_HCI_RAW_WARN`

Value:

```
if (ng_btsocket_hci_raw_debug_level >= NG_BTSOCKET_WARN_LEVEL) \
    printf
```

7.66.2 Function Documentation

7.66.2.1 `static LIST_HEAD (ng_btsocket_hci_raw_pcb) [static]`

Definition at line 117 of file `ng_btsocket_hci_raw.c`.

7.66.2.2 `void ng_btsocket_hci_raw_abort (struct socket * so)`

Definition at line 878 of file `ng_btsocket_hci_raw.c`.

7.66.2.3 `int ng_btsocket_hci_raw_attach (struct socket * so, int proto, struct thread * td)`

Definition at line 892 of file `ng_btsocket_hci_raw.c`.

References `BLUETOOTH_PROTO_HCI`, `M_NETGRAPH_BTSOCKET_HCI_RAW`, `ng_btsocket_hci_raw_node`, `NG_BTSOCKET_HCI_RAW_PRIVILEGED`, `NG_BTSOCKET_HCI_RAW_RECVSPACE`, `NG_BTSOCKET_HCI_RAW_SENDSPACE`, `NG_HCI_EVENT_COMMAND_COMPL`, `NG_HCI_EVENT_COMMAND_STATUS`, and `so2hci_raw_pcb`.

7.66.2.4 `int ng_btsocket_hci_raw_bind (struct socket * so, struct sockaddr * nam, struct thread * td)`

Definition at line 945 of file `ng_btsocket_hci_raw.c`.

References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_family`, `sockaddr_hci::hci_len`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

7.66.2.5 `void ng_btsocket_hci_raw_close (struct socket * so)`

Definition at line 883 of file `ng_btsocket_hci_raw.c`.

7.66.2.6 `int ng_btsocket_hci_raw_connect (struct socket * so, struct sockaddr * nam, struct thread * td)`

Definition at line 977 of file `ng_btsocket_hci_raw.c`.

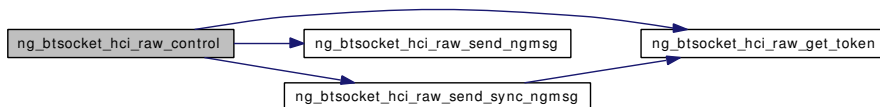
References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_family`, `sockaddr_hci::hci_len`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

7.66.2.7 `int ng_btsocket_hci_raw_control (struct socket * so, u_long cmd, caddr_t data, struct ifnet * ifp, struct thread * td)`

Definition at line 1016 of file `ng_btsocket_hci_raw.c`.

References `ng_btsocket_hci_raw_pcb::addr`, `ng_btsocket_hci_raw_node_bdaddr::bdaddr`, `ng_btsocket_hci_raw_node_buffer::buffer`, `ng_mesg::ng_msghdr::cmd`, `ng_btsocket_hci_raw_con_list::connections`, `ng_mesg::data`, `ng_btsocket_hci_raw_node_debug::debug`, `ng_btsocket_hci_raw_node_neighbor_cache::entries`, `ng_btsocket_hci_raw_node_features::features`, `ng_btsocket_hci_raw_pcb::flags`, `sockaddr_hci::hci_node`, `ng_mesg::header`, `min`, `ng_btsocket_hci_raw_pcb::msg`, `ng_btsocket_hci_raw_node_list_names::names`, `ng_btsocket_hci_raw_get_token()`, `ng_btsocket_hci_raw_node`, `NG_BT_SOCKET_HCI_RAW_PRIVILEGED`, `ng_btsocket_hci_raw_send_ngmsg()`, `ng_btsocket_hci_raw_send_sync_ngmsg()`, `NG_FREE_MSG`, `NG_HCI_MAX_CON_NUM`, `NG_HCI_MAX_NEIGHBOR_NUM`, `NG_HCI_NODE_TYPE`, `NG_MKMESSAGE`, `NG_NODESIZ`, `NG_SEND_MSG_PATH`, `NGM_GENERIC_COOKIE`, `NGM_HCI_COOKIE`, `NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE`, `NGM_HCI_NODE_GET_BDADDR`, `NGM_HCI_NODE_GET_BUFFER`, `NGM_HCI_NODE_GET_CON_LIST`, `NGM_HCI_NODE_GET_DEBUG`, `NGM_HCI_NODE_GET_FEATURES`, `NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK`, `NGM_HCI_NODE_GET_NEIGHBOR_CACHE`, `NGM_HCI_NODE_GET_PACKET_MASK`, `NGM_HCI_NODE_GET_ROLE_SWITCH`, `NGM_HCI_NODE_GET_STAT`, `NGM_HCI_NODE_GET_STATE`, `NGM_HCI_NODE_INIT`, `NGM_HCI_NODE_RESET_STAT`, `NGM_HCI_NODE_SET_DEBUG`, `NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK`, `NGM_HCI_NODE_SET_PACKET_MASK`, `NGM_HCI_NODE_SET_ROLE_SWITCH`, `NGM_LISTNAMES`, `namelist::nodeinfo`, `ng_hci_node_con_list_ep::num_connections`, `ng_btsocket_hci_raw_con_list::num_connections`, `ng_hci_node_get_neighbor_cache_ep::num_entries`, `ng_btsocket_hci_raw_node_neighbor_cache::num_entries`, `ng_btsocket_hci_raw_node_list_names::num_names`, `namelist::numnames`, `ng_btsocket_hci_raw_node_packet_mask::packet_mask`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, `ng_btsocket_hci_raw_node_link_policy_mask::policy_mask`, `ng_btsocket_hci_raw_node_role_switch::role_switch`, `SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_CACHE`, `SIOC_HCI_RAW_NODE_GET_BDADDR`, `SIOC_HCI_RAW_NODE_GET_BUFFER`, `SIOC_HCI_RAW_NODE_GET_CON_LIST`, `SIOC_HCI_RAW_NODE_GET_DEBUG`, `SIOC_HCI_RAW_NODE_GET_FEATURES`, `SIOC_HCI_RAW_NODE_GET_LINK_POLICY_MASK`, `SIOC_HCI_RAW_NODE_GET_NEIGHBOR_CACHE`, `SIOC_HCI_RAW_NODE_GET_PACKET_MASK`, `SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH`, `SIOC_HCI_RAW_NODE_GET_STAT`, `SIOC_HCI_RAW_NODE_GET_STATE`, `SIOC_HCI_RAW_NODE_INIT`, `SIOC_HCI_RAW_NODE_LIST_NAMES`, `SIOC_HCI_RAW_NODE_RESET_STAT`, `SIOC_HCI_RAW_NODE_SET_DEBUG`, `SIOC_HCI_RAW_NODE_SET_LINK_POLICY_MASK`, `SIOC_HCI_RAW_NODE_SET_PACKET_MASK`, `SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH`, `so2hci_raw_pcb`, `ng_btsocket_hci_raw_node_stat::stat`, `ng_btsocket_hci_raw_node_state::state`, `ng_mesg::ng_msghdr::token`, and `ng_btsocket_hci_raw_pcb::token`.

Here is the call graph for this function:



7.66.2.8 int ng_btsocket_hci_raw_ctloutput (struct socket * so, struct sockopt * sopt)

Definition at line 1393 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::filter, ng_btsocket_hci_raw_pcb::flags, NG_BT_SOCKET_HCI_RAW_DIRECTION, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, so2hci_raw_pcb, SO_HCI_RAW_DIRECTION, SO_HCI_RAW_FILTER, and SOL_HCI_RAW.

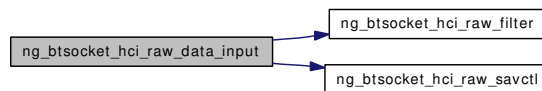
7.66.2.9 static void ng_btsocket_hci_raw_data_input (struct mbuf * nam) [static]

Definition at line 472 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::addr, sockaddr_hci::hci_node, ng_btsocket_hci_raw_filter(), NG_BT_SOCKET_HCI_RAW_INFO, ng_btsocket_hci_raw_savctl(), NG_FREE_M, ng_btsocket_hci_raw_pcb::pcb_mtx, and ng_btsocket_hci_raw_pcb::so.

Referenced by ng_btsocket_hci_raw_input().

Here is the call graph for this function:

**7.66.2.10 void ng_btsocket_hci_raw_detach (struct socket * so)**

Definition at line 1471 of file ng_btsocket_hci_raw.c.

References M_NETGRAPH_BT_SOCKET_HCI_RAW, ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

7.66.2.11 int ng_btsocket_hci_raw_disconnect (struct socket * so)

Definition at line 1501 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_node, ng_btsocket_hci_raw_pcb::pcb_mtx, and so2hci_raw_pcb.

7.66.2.12 static int ng_btsocket_hci_raw_filter (ng_btsocket_hci_raw_pcb_p, struct mbuf *, int) [static]

Definition at line 666 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::filter, ng_btsocket_hci_raw_pcb::flags, NG_BT_SOCKET_HCI_RAW_PRIVILEGED, NG_HCI_ACL_DATA_PKT, NG_HCI_CMD_PKT, NG_HCI_EVENT_PKT, NG_HCI_OCF, NG_HCI_OGF, NG_HCI_SCO_DATA_PKT, and ng_btsocket_hci_raw_pcb::pcb_mtx.

Referenced by ng_btsocket_hci_raw_data_input(), and ng_btsocket_hci_raw_send().

7.66.2.13 static void ng_btsocket_hci_raw_get_token (u_int32_t * token) [static]

Definition at line 359 of file ng_btsocket_hci_raw.c.

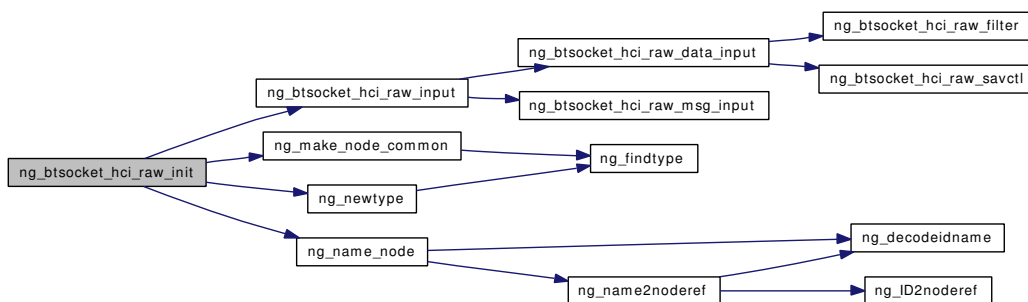
Referenced by ng_btsocket_hci_raw_control(), and ng_btsocket_hci_raw_send_sync_ngmsg().

7.66.2.14 void ng_btsocket_hci_raw_init (void)

Definition at line 721 of file ng_btsocket_hci_raw.c.

References ifqmaxlen, M_NETGRAPH_BT_SOCKET_HCI_RAW, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_HCI_RAW_ALERT, ng_btsocket_hci_raw_input(), ng_btsocket_hci_raw_node, NG_BT_SOCKET_HCI_RAW_NODE_TYPE, ng_btsocket_hci_raw_queue, ng_btsocket_hci_raw_queue_mtx, ng_btsocket_hci_raw_task, NG_HCI_EVENT_LINK_KEY_NOTIFICATION, NG_HCI_EVENT_RETURN_LINK_KEYS, NG_HCI_EVENT_VENDOR, NG_HCI_OCF_EXIT_PERIODIC_INQUIRY, NG_HCI_OCF_GET_LINK_QUALITY, NG_HCI_OCF_INQUIRY, NG_HCI_OCF_INQUIRY_CANCEL, NG_HCI_OCF_PERIODIC_INQUIRY, NG_HCI_OCF_READ_AUTH_ENABLE, NG_HCI_OCF_READ_AUTO_FLUSH_TIMO, NG_HCI_OCF_READ_BDADDR, NG_HCI_OCF_READ_BUFFER_SIZE, NG_HCI_OCF_READ_CLOCK_OFFSET, NG_HCI_OCF_READ_CON_ACCEPT_TIMO, NG_HCI_OCF_READ_COUNTRY_CODE, NG_HCI_OCF_READ_ENCRYPTION_MODE, NG_HCI_OCF_READ_FAILED_CONTACT_CNTR, NG_HCI_OCF_READ_HOLD_MODE_ACTIVITY, NG_HCI_OCF_READ_IAC_LAP, NG_HCI_OCF_READ_INQUIRY_SCAN_ACTIVITY, NG_HCI_OCF_READ_LINK_POLICY_SETTINGS, NG_HCI_OCF_READ_LINK_SUPERVISION_TIMO, NG_HCI_OCF_READ_LOCAL_FEATURES, NG_HCI_OCF_READ_LOCAL_NAME, NG_HCI_OCF_READ_LOCAL_VER, NG_HCI_OCF_READ_LOOPBACK_MODE, NG_HCI_OCF_READ_NUM_BROADCAST_RETRANS, NG_HCI_OCF_READ_PAGE_SCAN, NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY, NG_HCI_OCF_READ_PAGE_SCAN_PERIOD, NG_HCI_OCF_READ_PAGE_TIMO, NG_HCI_OCF_READ_PIN_TYPE, NG_HCI_OCF_READ_REMOTE_FEATURES, NG_HCI_OCF_READ_REMOTE_VER_INFO, NG_HCI_OCF_READ_RSSI, NG_HCI_OCF_READ_SCAN_ENABLE, NG_HCI_OCF_READ_SCO_FLOW_CONTROL, NG_HCI_OCF_READ_SUPPORTED_IAC_NUM, NG_HCI_OCF_READ_UNIT_CLASS, NG_HCI_OCF_READ_VOICE_SETTINGS, NG_HCI_OCF_READ_XMIT_LEVEL, NG_HCI_OCF_REMOTE_NAME_REQ, NG_HCI_OCF_ROLE_DISCOVERY, NG_HCI_OGF_HC_BASEBAND, NG_HCI_OGF_INFO, NG_HCI_OGF_LINK_CONTROL, NG_HCI_OGF_LINK_POLICY, NG_HCI_OGF_STATUS, NG_HCI_OGF_TESTING, ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and tpestruct.

Here is the call graph for this function:



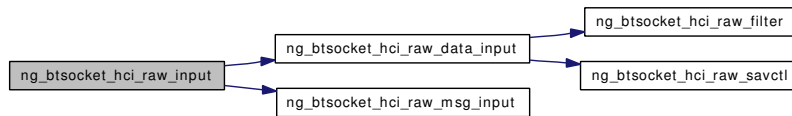
7.66.2.15 static void ng_btsocket_hci_raw_input (void *, int) [static]

Definition at line 581 of file ng_btsocket_hci_raw.c.

References ng_item::el_flags, NG_BT_ITEMQ_DEQUEUE, ng_btsocket_hci_raw_data_input(), ng_btsocket_hci_raw_msg_input(), ng_btsocket_hci_raw_queue, ng_btsocket_hci_raw_queue_mtx, NG_FREE_ITEM, NGL_GET_M, NGL_GET_MSG, NGQF_DATA, NGQF_MESG, and NGQF_TYPE.

Referenced by ng_btsocket_hci_raw_init().

Here is the call graph for this function:



7.66.2.16 static void ng_btsocket_hci_raw_msg_input (struct ng_mesg * msg) [static]

Definition at line 549 of file ng_btsocket_hci_raw.c.

References ng_mesg::header, ng_btsocket_hci_raw_pcb::msg, NG_FREE_MSG, ng_btsocket_hci_raw_pcb::pcb_mtx, ng_btsocket_hci_raw_pcb::token, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_hci_raw_input().

7.66.2.17 static int ng_btsocket_hci_raw_node_connect (hook_p hook) [static]

Definition at line 227 of file ng_btsocket_hci_raw.c.

7.66.2.18 static int ng_btsocket_hci_raw_node_disconnect (hook_p hook) [static]

Definition at line 237 of file ng_btsocket_hci_raw.c.

7.66.2.19 static int ng_btsocket_hci_raw_node_newhook (node_p node, hook_p hook, char const * name) [static]

Definition at line 217 of file ng_btsocket_hci_raw.c.

7.66.2.20 static int ng_btsocket_hci_raw_node_rcvdata (hook_p hook, item_p item) [static]

Definition at line 297 of file ng_btsocket_hci_raw.c.

References sockaddr_hci::hci_family, sockaddr_hci::hci_len, sockaddr_hci::hci_node, NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_SOCKET_HCI_RAW_ERR, ng_btsocket_hci_raw_queue, ng_btsocket_hci_raw_queue_mtx, ng_btsocket_hci_raw_wakeup_input_task, NG_FREE_ITEM, NG_PEER_NODE_NAME, NGL_GET_M, and NGL_M.

7.66.2.21 static int ng_btsocket_hci_raw_node_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 249 of file ng_btsocket_hci_raw.c.

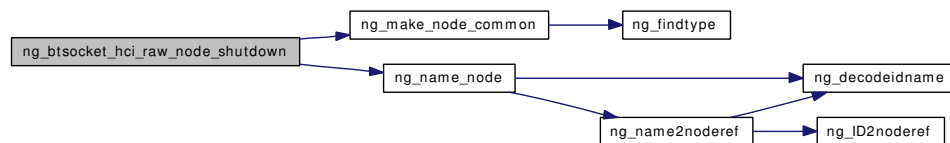
References ng_mesg::ng_msghdr::flags, ng_mesg::header, NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_SOCKET_HCI_RAW_ERR, ng_btsocket_hci_raw_queue, ng_btsocket_hci_raw_queue_mtx, ng_btsocket_hci_raw_wakeup_input_task, NG_FREE_ITEM, NGF_RESP, NGL_MSG, NGM_GENERIC_COOKIE, NGM_HCI_COOKIE, ng_mesg::ng_msghdr::token, and ng_mesg::ng_msghdr::typecookie.

7.66.2.22 static int ng_btsocket_hci_raw_node_shutdown (node_p node) [static]

Definition at line 181 of file ng_btsocket_hci_raw.c.

References NG_BTSTACK_HCI_RAW_ALERT, ng_btsocket_hci_raw_node, NG_BTSTACK_HCI_RAW_NODE_TYPE, ng_make_node_common(), ng_name_node(), NG_NODE_UNREF, and tpestruct.

Here is the call graph for this function:



7.66.2.23 static void ng_btsocket_hci_raw_output (node_p, hook_p, void *, int) [static]

Definition at line 623 of file ng_btsocket_hci_raw.c.

References sockaddr_hci::hci_node, NG_FREE_M, NG_HOOK_NOT_VALID, NG_NODE_NOT_VALID, NG_PEER_NODE, NG_PEER_NODE_NAME, and NG_SEND_DATA_ONLY.

Referenced by ng_btsocket_hci_raw_send().

7.66.2.24 int ng_btsocket_hci_raw_peeraddr (struct socket * so, struct sockaddr ** nam)

Definition at line 1522 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_sockaddr().

Here is the call graph for this function:



7.66.2.25 static void ng_btsocket_hci_raw_savctl (ng_btsocket_hci_raw_pcb_p, struct mbuf **, struct mbuf *) [static]

Definition at line 442 of file ng_btsocket_hci_raw.c.

References ng_btsocket_hci_raw_pcb::flags, NG_BTSTACK_HCI_RAW_DIRECTION, ng_btsocket_hci_raw_pcb::pcb_mtx, SCM_HCI_RAW_DIRECTION, ng_btsocket_hci_raw_pcb::so, and SOL_HCI_RAW.

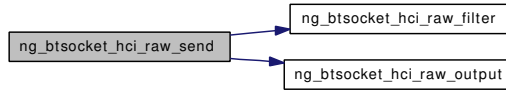
Referenced by ng_btsocket_hci_raw_data_input().

7.66.2.26 int ng_btsocket_hci_raw_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * sa, struct mbuf * control, struct thread * td)

Definition at line 1532 of file ng_btsocket_hci_raw.c.

References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_node`, `ng_btsocket_hci_raw_filter()`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_output()`, `NG_FREE_M`, `NG_HCI_CMD_PKT`, `NG_HCI_CMD_PKT_SIZE`, `ng_send_fn`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

Here is the call graph for this function:



7.66.2.27 `static int ng_btsocket_hci_raw_send_ngmsg (char * path, int cmd, void * arg, int arglen)` [static]

Definition at line 376 of file `ng_btsocket_hci_raw.c`.

References `ng_mesg::data`, `ng_btsocket_hci_raw_node`, `NG_MKMESSAGE`, `NG_SEND_MSG_PATH`, and `NGM_HCI_COOKIE`.

Referenced by `ng_btsocket_hci_raw_control()`.

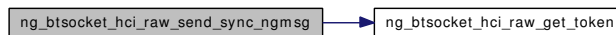
7.66.2.28 `static int ng_btsocket_hci_raw_send_sync_ngmsg (ng_btsocket_hci_raw_pcb_p pcb, char * path, int cmd, void * rsp, int rsplen)` [static]

Definition at line 398 of file `ng_btsocket_hci_raw.c`.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `ng_btsocket_hci_raw_pcb::msg`, `ng_btsocket_hci_raw_get_token()`, `ng_btsocket_hci_raw_node`, `NG_FREE_MSG`, `NG_MKMESSAGE`, `NG_SEND_MSG_PATH`, `NGM_HCI_COOKIE`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, `ng_btsocket_hci_raw_pcb::token`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_btsocket_hci_raw_control()`.

Here is the call graph for this function:



7.66.2.29 `int ng_btsocket_hci_raw_sockaddr (struct socket * so, struct sockaddr ** nam)`

Definition at line 1617 of file `ng_btsocket_hci_raw.c`.

References `ng_btsocket_hci_raw_pcb::addr`, `sockaddr_hci::hci_node`, `ng_btsocket_hci_raw_node`, `ng_btsocket_hci_raw_pcb::pcb_mtx`, and `so2hci_raw_pcb`.

Referenced by `ng_btsocket_hci_raw_peeraddr()`.

7.66.3 Variable Documentation

7.66.3.1 `int ifqmaxlen`

Referenced by `ng_btsocket_hci_raw_init()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_rfcomm_session_create()`, and `ng_device_constructor()`.

7.66.3.2 [u_int32_t ng_btsocket_hci_raw_debug_level](#) [static]

Definition at line 111 of file ng_btsocket_hci_raw.c.

7.66.3.3 [u_int32_t ng_btsocket_hci_raw_ioctl_timeout](#) [static]

Definition at line 112 of file ng_btsocket_hci_raw.c.

7.66.3.4 [node_p ng_btsocket_hci_raw_node](#) [static]

Definition at line 113 of file ng_btsocket_hci_raw.c.

Referenced by [ng_btsocket_hci_raw_attach\(\)](#), [ng_btsocket_hci_raw_bind\(\)](#), [ng_btsocket_hci_raw_connect\(\)](#), [ng_btsocket_hci_raw_control\(\)](#), [ng_btsocket_hci_raw_ctloutput\(\)](#), [ng_btsocket_hci_raw_detach\(\)](#), [ng_btsocket_hci_raw_disconnect\(\)](#), [ng_btsocket_hci_raw_init\(\)](#), [ng_btsocket_hci_raw_node_shutdown\(\)](#), [ng_btsocket_hci_raw_send\(\)](#), [ng_btsocket_hci_raw_send_ngmsg\(\)](#), [ng_btsocket_hci_raw_send_sync_ngmsg\(\)](#), and [ng_btsocket_hci_raw_sockaddr\(\)](#).

7.66.3.5 [ng_connect_t ng_btsocket_hci_raw_node_connect](#) [static]

Definition at line 75 of file ng_btsocket_hci_raw.c.

7.66.3.6 [ng_constructor_t ng_btsocket_hci_raw_node_constructor](#) [static]

Definition at line 71 of file ng_btsocket_hci_raw.c.

7.66.3.7 [ng_disconnect_t ng_btsocket_hci_raw_node_disconnect](#) [static]

Definition at line 77 of file ng_btsocket_hci_raw.c.

7.66.3.8 [ng_newhook_t ng_btsocket_hci_raw_node_newhook](#) [static]

Definition at line 74 of file ng_btsocket_hci_raw.c.

7.66.3.9 [ng_rcvdata_t ng_btsocket_hci_raw_node_rcvdata](#) [static]

Definition at line 76 of file ng_btsocket_hci_raw.c.

7.66.3.10 [ng_rcvmsg_t ng_btsocket_hci_raw_node_rcvmsg](#) [static]

Definition at line 72 of file ng_btsocket_hci_raw.c.

7.66.3.11 [ng_shutdown_t ng_btsocket_hci_raw_node_shutdown](#) [static]

Definition at line 73 of file ng_btsocket_hci_raw.c.

7.66.3.12 struct `ng_bt_itemq ng_btsocket_hci_raw_queue` [static]

Definition at line 114 of file `ng_btsocket_hci_raw.c`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_hci_raw_node_rcvdata()`, and `ng_btsocket_hci_raw_node_rcvmsg()`.

7.66.3.13 struct `mtx ng_btsocket_hci_raw_queue_mtx` [static]

Definition at line 115 of file `ng_btsocket_hci_raw.c`.

Referenced by `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_hci_raw_node_rcvdata()`, and `ng_btsocket_hci_raw_node_rcvmsg()`.

7.66.3.14 struct `task ng_btsocket_hci_raw_task` [static]

Definition at line 116 of file `ng_btsocket_hci_raw.c`.

Referenced by `ng_btsocket_hci_raw_init()`.

7.66.3.15 struct `ng_type typestruct` [static]**Initial value:**

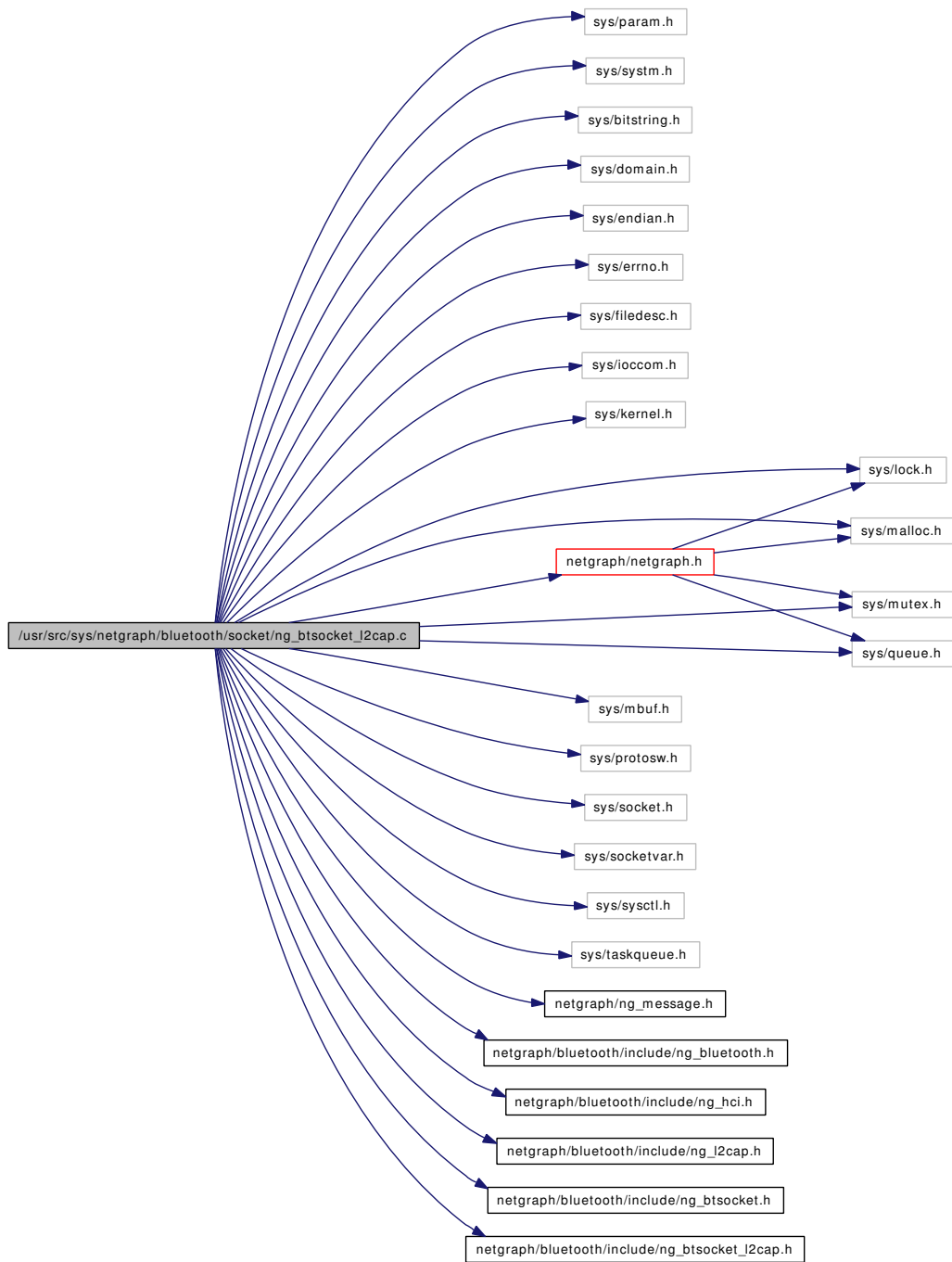
```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_BT_SOCKET_HCI_RAW_NODE_TYPE,
    .constructor = ng_btsocket_hci_raw_node_constructor,
    .rcvmsg =      ng_btsocket_hci_raw_node_rcvmsg,
    .shutdown =    ng_btsocket_hci_raw_node_shutdown,
    .newhook =     ng_btsocket_hci_raw_node_newhook,
    .connect =     ng_btsocket_hci_raw_node_connect,
    .rcvdata =     ng_btsocket_hci_raw_node_rcvdata,
    .disconnect =  ng_btsocket_hci_raw_node_disconnect,
}
```

Definition at line 97 of file `ng_btsocket_hci_raw.c`.

7.67 /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bitstring.h>
#include <sys/domain.h>
#include <sys/endian.h>
#include <sys/errno.h>
#include <sys/filedesc.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/protosw.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/taskqueue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket.h>
#include <netgraph/bluetooth/include/ng_btsocket_l2cap.h>
```

Include dependency graph for ng_btsocket_l2cap.c:



Defines

- #define [M_NETGRAPH_BT_SOCKET_L2CAP](#) M_NETGRAPH
- #define [NG_BT_SOCKET_L2CAP_INFO](#)
- #define [NG_BT_SOCKET_L2CAP_WARN](#)
- #define [NG_BT_SOCKET_L2CAP_ERR](#)
- #define [NG_BT_SOCKET_L2CAP_ALERT](#)

- #define `ng_btsocket_l2cap_wakeup_input_task()` `taskqueue_enqueue(taskqueue_swi_giant, &ng_btsocket_l2cap_queue_task)`
- #define `ng_btsocket_l2cap_wakeup_route_task()` `taskqueue_enqueue(taskqueue_swi_giant, &ng_btsocket_l2cap_rt_task)`

Functions

- static void `ng_btsocket_l2cap_input` (void *, int)
- static void `ng_btsocket_l2cap_rtclean` (void *, int)
- static LIST_HEAD (`ng_btsocket_l2cap_pcb`)
- static int `ng_btsocket_l2cap_node_shutdown` (`node_p` node)
- static int `ng_btsocket_l2cap_node_newhook` (`node_p` node, `hook_p` hook, char const *name)
- static int `ng_btsocket_l2cap_node_connect` (`hook_p` hook)
- static int `ng_btsocket_l2cap_node_disconnect` (`hook_p` hook)
- static int `ng_btsocket_l2cap_node_rcvmsg` (`node_p` node, `item_p` item, `hook_p` hook)
- static int `ng_btsocket_l2cap_node_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_btsocket_l2cap_process_l2ca_con_req_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_con_ind` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_cfg_ind` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_discon_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_discon_ind` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_process_l2ca_write_rsp` (struct `ng_mesg` *msg, `ng_btsocket_l2cap_rentry_p` rt)
- static int `ng_btsocket_l2cap_send_l2ca_con_req` (`ng_btsocket_l2cap_pcb_p` pcb)
- static int `ng_btsocket_l2cap_send_l2ca_con_rsp_req` (u_int32_t token, `ng_btsocket_l2cap_rentry_p` rt, `bdaddr_p` dst, int ident, int lcid, int result)
- static int `ng_btsocket_l2cap_send_l2ca_cfg_req` (`ng_btsocket_l2cap_pcb_p` pcb)
- static int `ng_btsocket_l2cap_send_l2ca_cfg_rsp` (`ng_btsocket_l2cap_pcb_p` pcb)
- static int `ng_btsocket_l2cap_send_l2ca_discon_req` (u_int32_t token, `ng_btsocket_l2cap_pcb_p` pcb)
- static void `ng_btsocket_l2cap_data_input` (struct `mbuf` *m, `hook_p` hook)
- static void `ng_btsocket_l2cap_default_msg_input` (struct `ng_mesg` *msg, `hook_p` hook)
- static void `ng_btsocket_l2cap_l2ca_msg_input` (struct `ng_mesg` *msg, `hook_p` hook)
- void `ng_btsocket_l2cap_init` (void)
- void `ng_btsocket_l2cap_abort` (struct `socket` *so)
- void `ng_btsocket_l2cap_close` (struct `socket` *so)
- int `ng_btsocket_l2cap_accept` (struct `socket` *so, struct `sockaddr` **nam)
- int `ng_btsocket_l2cap_attach` (struct `socket` *so, int proto, struct `thread` *td)
- int `ng_btsocket_l2cap_bind` (struct `socket` *so, struct `sockaddr` *nam, struct `thread` *td)

- int `ng_btsocket_l2cap_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- int `ng_btsocket_l2cap_control` (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- int `ng_btsocket_l2cap_ctloutput` (struct socket *so, struct sockopt *sopt)
- void `ng_btsocket_l2cap_detach` (struct socket *so)
- int `ng_btsocket_l2cap_disconnect` (struct socket *so)
- int `ng_btsocket_l2cap_listen` (struct socket *so, int backlog, struct thread *td)
- int `ng_btsocket_l2cap_peeraddr` (struct socket *so, struct sockaddr **nam)
- int `ng_btsocket_l2cap_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *nam, struct mbuf *control, struct thread *td)
- static int `ng_btsocket_l2cap_send2` (ng_btsocket_l2cap_pcb_p pcb)
- int `ng_btsocket_l2cap_sockaddr` (struct socket *so, struct sockaddr **nam)
- static `ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_addr` (bdaddr_p bdaddr, int psm)
- static `ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_token` (u_int32_t token)
- static `ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_cid` (bdaddr_p src, int cid)
- static void `ng_btsocket_l2cap_timeout` (ng_btsocket_l2cap_pcb_p pcb)
- static void `ng_btsocket_l2cap_untimeout` (ng_btsocket_l2cap_pcb_p pcb)
- static void `ng_btsocket_l2cap_process_timeout` (void *xpcb)
- static int `ng_btsocket_l2cap_result2errno` (int result)

Variables

- static `ng_constructor_t ng_btsocket_l2cap_node_constructor`
- static `ng_rcvmsg_t ng_btsocket_l2cap_node_rcvmsg`
- static `ng_shutdown_t ng_btsocket_l2cap_node_shutdown`
- static `ng_newhook_t ng_btsocket_l2cap_node_newhook`
- static `ng_connect_t ng_btsocket_l2cap_node_connect`
- static `ng_rcvdata_t ng_btsocket_l2cap_node_rcvdata`
- static `ng_disconnect_t ng_btsocket_l2cap_node_disconnect`
- static struct `ng_type` typestruct
- int `ifqmaxlen`
- static u_int32_t `ng_btsocket_l2cap_debug_level`
- static `node_p ng_btsocket_l2cap_node`
- static struct `ng_bt_itemq ng_btsocket_l2cap_queue`
- static struct mtx `ng_btsocket_l2cap_queue_mtx`
- static struct task `ng_btsocket_l2cap_queue_task`

7.67.1 Define Documentation

7.67.1.1 #define M_NETGRAPH_BT SOCKET_L2CAP M_NETGRAPH

Definition at line 66 of file `ng_btsocket_l2cap.c`.

Referenced by `ng_btsocket_l2cap_attach()`, `ng_btsocket_l2cap_default_msg_input()`, and `ng_btsocket_l2cap_detach()`.

7.67.1.2 #define NG_BT_SOCKET_L2CAP_ALERT

Value:

```
if (ng_btsocket_l2cap_debug_level >= NG_BT_SOCKET_ALERT_LEVEL) \  
    printf
```

Referenced by `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_l2ca_msg_input()`, and `ng_btsocket_l2cap_node_shutdown()`.

7.67.1.3 #define NG_BT_SOCKET_L2CAP_ERR

Value:

```
if (ng_btsocket_l2cap_debug_level >= NG_BT_SOCKET_ERR_LEVEL) \  
    printf
```

Referenced by `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_node_rcvdata()`, `ng_btsocket_l2cap_node_rcvmsg()`, `ng_btsocket_l2cap_process_timeout()`, `ng_btsocket_l2cap_send()`, and `ng_btsocket_l2cap_send2()`.

7.67.1.4 #define NG_BT_SOCKET_L2CAP_INFO

Value:

```
if (ng_btsocket_l2cap_debug_level >= NG_BT_SOCKET_INFO_LEVEL) \  
    printf
```

Referenced by `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_ind()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, and `ng_btsocket_l2cap_send2()`.

7.67.1.5 #define ng_btsocket_l2cap_wakeup_input_task() taskqueue_enqueue(taskqueue_swigiant, &ng_btsocket_l2cap_queue_task)

Referenced by `ng_btsocket_l2cap_node_rcvdata()`, and `ng_btsocket_l2cap_node_rcvmsg()`.

7.67.1.6 #define ng_btsocket_l2cap_wakeup_route_task() taskqueue_enqueue(taskqueue_swigiant, &ng_btsocket_l2cap_rt_task)

Referenced by `ng_btsocket_l2cap_node_disconnect()`.

7.67.1.7 #define NG_BT_SOCKET_L2CAP_WARN

Value:

```
if (ng_btsocket_l2cap_debug_level >= NG_BT_SOCKET_WARN_LEVEL) \  
    printf
```

Referenced by `ng_btsocket_l2cap_default_msg_input()`, and `ng_btsocket_l2cap_l2ca_msg_input()`.

7.67.2 Function Documentation

7.67.2.1 static LIST_HEAD (ng_btsocket_l2cap_pcb) [static]

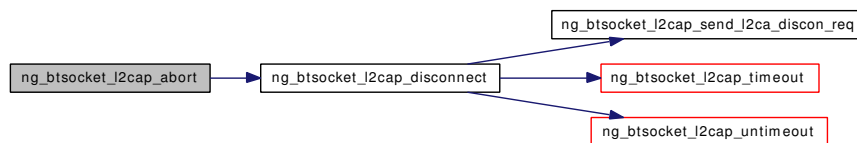
Definition at line 101 of file ng_btsocket_l2cap.c.

7.67.2.2 void ng_btsocket_l2cap_abort (struct socket * so)

Definition at line 1864 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_disconnect().

Here is the call graph for this function:



7.67.2.3 int ng_btsocket_l2cap_accept (struct socket * so, struct sockaddr ** nam)

Definition at line 1884 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_node, and ng_btsocket_l2cap_peeraddr().

Here is the call graph for this function:



7.67.2.4 int ng_btsocket_l2cap_attach (struct socket * so, int proto, struct thread * td)

Definition at line 1897 of file ng_btsocket_l2cap.c.

References BLUETOOTH_PROTO_L2CAP, M_NETGRAPH_BT SOCKET_L2CAP, NG_BT SOCKET_L2CAP_CLOSED, ng_btsocket_l2cap_node, NG_BT SOCKET_L2CAP_RECVSPACE, NG_BT SOCKET_L2CAP_SENDSPACE, NG_HCI_SERVICE_TYPE_BEST_EFFORT, NG_L2CAP_FLUSH_TIMO_DEFAULT, NG_L2CAP_LINK_TIMO_DEFAULT, NG_L2CAP_MTU_DEFAULT, and so2l2cap_pcb.

7.67.2.5 int ng_btsocket_l2cap_bind (struct socket * so, struct sockaddr * nam, struct thread * td)

Definition at line 2009 of file ng_btsocket_l2cap.c.

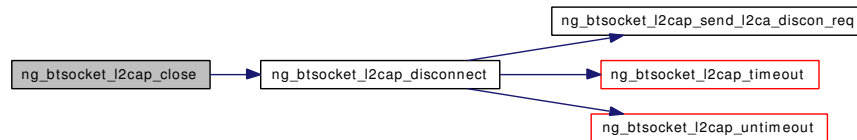
References sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::src.

7.67.2.6 void ng_btsocket_l2cap_close (struct socket * so)

Definition at line 1872 of file ng_btsocket_l2cap.c.

References `ng_btsocket_l2cap_disconnect()`.

Here is the call graph for this function:

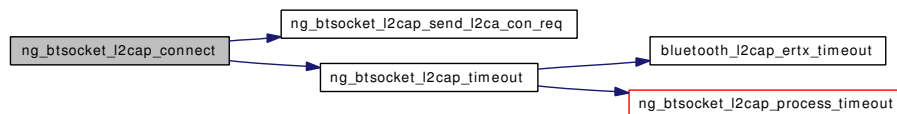


7.67.2.7 `int ng_btsocket_l2cap_connect (struct socket * so, struct sockaddr * nam, struct thread * td)`

Definition at line 2065 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::dst`, `ng_btsocket_l2cap_pcb::flags`, `ng_btsocket_l2cap_rtrtry::hook`, `sockaddr_l2cap::l2cap_family`, `sockaddr_l2cap::l2cap_len`, `NG_BT_SOCKET_L2CAP_CLIENT`, `NG_BT_SOCKET_L2CAP_CONNECTING`, `ng_btsocket_l2cap_node`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_timeout()`, `NG_HCI_BDADDR_ANY`, `NG_HOOK_NOT_VALID`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::psm`, `ng_btsocket_l2cap_pcb::rt`, `ng_btsocket_l2cap_pcb::so`, `so2l2cap_pcb`, `ng_btsocket_l2cap_rtrtry::src`, `ng_btsocket_l2cap_pcb::src`, and `ng_btsocket_l2cap_pcb::state`.

Here is the call graph for this function:



7.67.2.8 `int ng_btsocket_l2cap_control (struct socket * so, u_long cmd, caddr_t data, struct ifnet * ifp, struct thread * td)`

Definition at line 2161 of file `ng_btsocket_l2cap.c`.

7.67.2.9 `int ng_btsocket_l2cap_ctloutput (struct socket * so, struct sockopt * sopt)`

Definition at line 2172 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flush_timo`, `ng_btsocket_l2cap_pcb::iflow`, `ng_btsocket_l2cap_pcb::imtu`, `NG_BT_SOCKET_L2CAP_CLOSED`, `ng_btsocket_l2cap_node`, `ng_btsocket_l2cap_pcb::oflow`, `ng_btsocket_l2cap_pcb::omtu`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `so2l2cap_pcb`, `SO_L2CAP_FLUSH`, `SO_L2CAP_IFLOW`, `SO_L2CAP_IMTU`, `SO_L2CAP_OFLOW`, `SO_L2CAP_OMTU`, `SOL_L2CAP`, and `ng_btsocket_l2cap_pcb::state`.

7.67.2.10 `static void ng_btsocket_l2cap_data_input (struct mbuf * m, hook_p hook)` [static]

Definition at line 1323 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::imtu`, `NG_BT_SOCKET_L2CAP_ALERT`, `NG_BT_SOCKET_L2CAP_ERR`, `NG_BT_SOCKET_L2CAP_INFO`, `NG_BT_SOCKET_L2CAP_OPEN`, `ng_btsocket_l2cap_pcb_by_cid()`, `NG_FREE_M`, `NG_HOOK_PRIVATE`, `NG_L2CAP_CLT_CID`, `NG_L2CAP_FIRST_CID`, `NG_L2CAP_MTU_DEFAULT`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::psm`, `ng_btsocket_l2cap_pcb::so`, `ng_btsocket_l2cap_pcb::src`, `ng_btsocket_l2cap_rtenry::src`, and `ng_btsocket_l2cap_pcb::state`.

Referenced by `ng_btsocket_l2cap_input()`.

Here is the call graph for this function:



7.67.2.11 `static void ng_btsocket_l2cap_default_msg_input (struct ng_mesg * msg, hook_p hook)` [static]

Definition at line 1544 of file `ng_btsocket_l2cap.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `ng_btsocket_l2cap_rtenry::hook`, `M_NETGRAPH_BT_SOCKET_L2CAP`, `NG_BT_SOCKET_L2CAP_INFO`, `NG_BT_SOCKET_L2CAP_WARN`, `NG_FREE_MSG`, `NG_HCI_BDADDR_ANY`, `NG_HOOK_NAME`, `NG_HOOK_PRIVATE`, `NG_HOOK_SET_PRIVATE`, `NGM_L2CAP_NODE_HOOK_INFO`, and `ng_btsocket_l2cap_rtenry::src`.

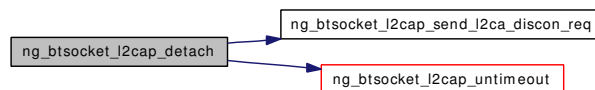
Referenced by `ng_btsocket_l2cap_input()`.

7.67.2.12 `void ng_btsocket_l2cap_detach (struct socket * so)`

Definition at line 2275 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flags`, `M_NETGRAPH_BT_SOCKET_L2CAP`, `NG_BT_SOCKET_L2CAP_CLOSED`, `NG_BT_SOCKET_L2CAP_DISCONNECTING`, `ng_btsocket_l2cap_node`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `NG_BT_SOCKET_L2CAP_TIMO`, `ng_btsocket_l2cap_untimeout()`, `ng_btsocket_l2cap_pcb::pcb_mtx`, and `so2l2cap_pcb`.

Here is the call graph for this function:



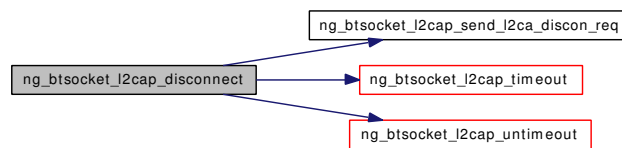
7.67.2.13 `int ng_btsocket_l2cap_disconnect (struct socket * so)`

Definition at line 2316 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flags`, `NG_BT_SOCKET_L2CAP_CLOSED`, `NG_BT_SOCKET_L2CAP_DISCONNECTING`, `ng_btsocket_l2cap_node`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_btsocket_l2cap_timeout()`, `NG_BT_SOCKET_L2CAP_TIMO`, `ng_btsocket_l2cap_untimeout()`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `so2l2cap_pcb`, and `ng_btsocket_l2cap_pcb::state`.

Referenced by `ng_btsocket_l2cap_abort()`, and `ng_btsocket_l2cap_close()`.

Here is the call graph for this function:

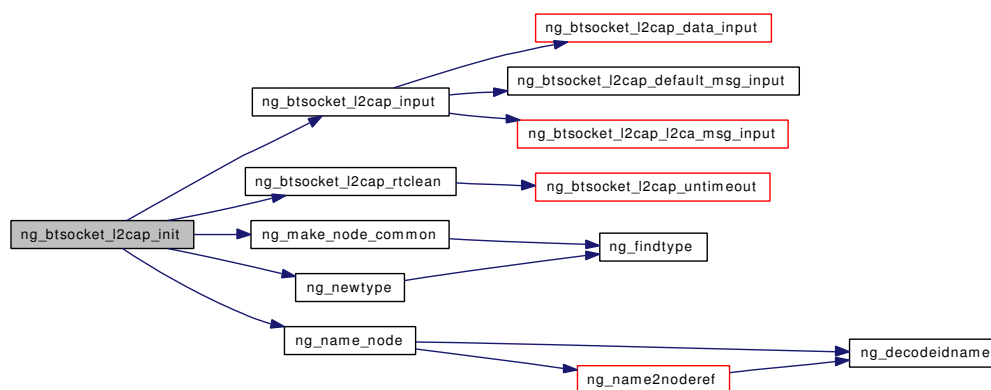


7.67.2.14 void ng_btsocket_l2cap_init (void)

Definition at line 1800 of file ng_btsocket_l2cap.c.

References ifqmaxlen, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_L2CAP_ALERT, ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node, NG_BT_SOCKET_L2CAP_NODE_TYPE, ng_btsocket_l2cap_queue, ng_btsocket_l2cap_queue_mtx, ng_btsocket_l2cap_queue_task, ng_btsocket_l2cap_rtclean(), ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



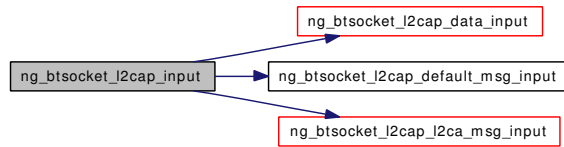
7.67.2.15 static void ng_btsocket_l2cap_input (void *, int) [static]

Definition at line 1668 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::cmd, ng_item::el_flags, ng_mesg::header, NG_BT_ITEMQ_DEQUEUE, ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_queue, ng_btsocket_l2cap_queue_mtx, NG_FREE_ITEM, NG_HOOK_NOT_VALID, NG_HOOK_UNREF, NGI_GET_HOOK, NGI_GET_M, NGI_GET_MSG, NGM_L2CAP_L2CA_CFG, NGM_L2CAP_L2CA_CFG_IND, NGM_L2CAP_L2CA_CFG_RSP, NGM_L2CAP_L2CA_CON, NGM_L2CAP_L2CA_CON_IND, NGM_L2CAP_L2CA_CON_RSP, NGM_L2CAP_L2CA_DISCON, NGM_L2CAP_L2CA_DISCON_IND, NGM_L2CAP_L2CA_WRITE, NGQF_DATA, NGQF_MESG, and NGQF_TYPE.

Referenced by ng_btsocket_l2cap_init().

Here is the call graph for this function:



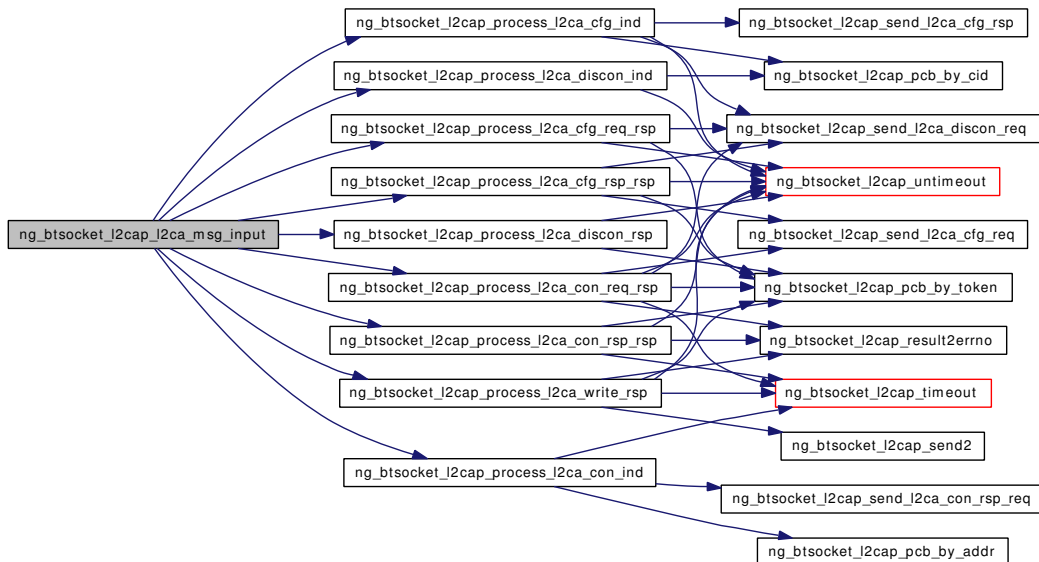
7.67.2.16 static void ng_btsocket_l2cap_l2ca_msg_input (struct ng_mesg * msg, hook_p hook) [static]

Definition at line 1598 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::cmd, ng_mesg::header, NG_BT_SOCKET_L2CAP_ALERT, ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_discon_ind(), ng_btsocket_l2cap_process_l2ca_discon_rsp(), ng_btsocket_l2cap_process_l2ca_write_rsp(), NG_BT_SOCKET_L2CAP_WARN, NG_FREE_MSG, NG_HOOK_PRIVATE, NGM_L2CAP_L2CA_CFG, NGM_L2CAP_L2CA_CFG_IND, NGM_L2CAP_L2CA_CFG_RSP, NGM_L2CAP_L2CA_CON, NGM_L2CAP_L2CA_CON_IND, NGM_L2CAP_L2CA_CON_RSP, NGM_L2CAP_L2CA_DISCON, NGM_L2CAP_L2CA_DISCON_IND, and NGM_L2CAP_L2CA_WRITE.

Referenced by ng_btsocket_l2cap_input().

Here is the call graph for this function:



7.67.2.17 int ng_btsocket_l2cap_listen (struct socket * so, int backlog, struct thread * td)

Definition at line 2359 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, and so2l2cap_pcb.

7.67.2.18 `static int ng_btsocket_l2cap_node_connect (hook_p hook)` [static]

Definition at line 280 of file ng_btsocket_l2cap.c.

References NG_HOOK_FORCE_QUEUE, NG_HOOK_PEER, NG_HOOK_REF, and NG_HOOK_SET_PRIVATE.

7.67.2.19 `static int ng_btsocket_l2cap_node_disconnect (hook_p hook)` [static]

Definition at line 298 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_wakeup_route_task, NG_HOOK_PRIVATE, and NG_HOOK_UNREF.

7.67.2.20 `static int ng_btsocket_l2cap_node_newhook (node_p node, hook_p hook, char const * name)` [static]

Definition at line 270 of file ng_btsocket_l2cap.c.

7.67.2.21 `static int ng_btsocket_l2cap_node_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 358 of file ng_btsocket_l2cap.c.

References NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_SOCKET_L2CAP_ERR, ng_btsocket_l2cap_queue, ng_btsocket_l2cap_queue_mtx, ng_btsocket_l2cap_wakeup_input_task, NG_FREE_ITEM, NG_HOOK_REF, and NGL_SET_HOOK.

7.67.2.22 `static int ng_btsocket_l2cap_node_rcvmsg (node_p node, item_p item, hook_p hook)` [static]

Definition at line 321 of file ng_btsocket_l2cap.c.

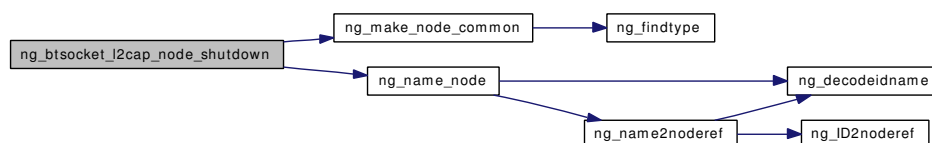
References ng_mesg::header, NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_SOCKET_L2CAP_ERR, ng_btsocket_l2cap_queue, ng_btsocket_l2cap_queue_mtx, ng_btsocket_l2cap_wakeup_input_task, NG_FREE_ITEM, NG_HOOK_REF, NGL_MSG, NGL_SET_HOOK, NGM_L2CAP_COOKIE, and ng_mesg::ng_msghdr::typecookie.

7.67.2.23 `static int ng_btsocket_l2cap_node_shutdown (node_p node)` [static]

Definition at line 233 of file ng_btsocket_l2cap.c.

References NG_BT_SOCKET_L2CAP_ALERT, ng_btsocket_l2cap_node, NG_BT_SOCKET_L2CAP_NODE_TYPE, ng_make_node_common(), ng_name_node(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



7.67.2.24 `static ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_addr (bdaddr_p bdaddr, int psm)` [static]

Definition at line 2576 of file `ng_btsocket_l2cap.c`.

References `NG_HCI_BDADDR_ANY`, `ng_btsocket_l2cap_pcb::psm`, `ng_btsocket_l2cap_pcb::so`, and `ng_btsocket_l2cap_pcb::src`.

Referenced by `ng_btsocket_l2cap_process_l2ca_con_ind()`.

7.67.2.25 `static ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_cid (bdaddr_p src, int cid)` [static]

Definition at line 2625 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::cid`, and `ng_btsocket_l2cap_pcb::src`.

Referenced by `ng_btsocket_l2cap_data_input()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, and `ng_btsocket_l2cap_process_l2ca_discon_ind()`.

7.67.2.26 `static ng_btsocket_l2cap_pcb_p ng_btsocket_l2cap_pcb_by_token (u_int32_t token)` [static]

Definition at line 2603 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::token`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, and `ng_btsocket_l2cap_process_l2ca_write_rsp()`.

7.67.2.27 `int ng_btsocket_l2cap_peeraddr (struct socket * so, struct sockaddr ** nam)`

Definition at line 2391 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::dst`, `sockaddr_l2cap::l2cap_bdaddr`, `sockaddr_l2cap::l2cap_family`, `sockaddr_l2cap::l2cap_len`, `sockaddr_l2cap::l2cap_psm`, `ng_btsocket_l2cap_node`, `ng_btsocket_l2cap_pcb::psm`, and `so_l2cap_pcb`.

Referenced by `ng_btsocket_l2cap_accept()`.

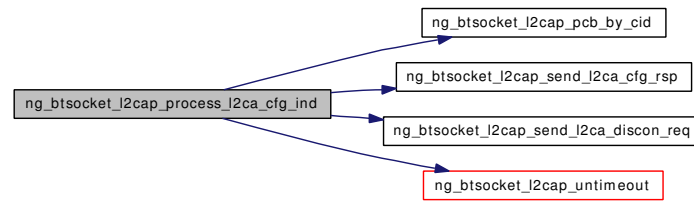
7.67.2.28 `static int ng_btsocket_l2cap_process_l2ca_cfg_ind (struct ng_mesg * msg, ng_btsocket_l2cap_rteentry_p rt)` [static]

Definition at line 875 of file `ng_btsocket_l2cap.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_btsocket_l2cap_pcb::cfg_state`, `ng_btsocket_l2cap_pcb::cid`, `ng_mesg::data`, `ng_btsocket_l2cap_pcb::dst`, `ng_btsocket_l2cap_pcb::flush_timo`, `ng_mesg::header`, `ng_btsocket_l2cap_pcb::iflow`, `NG_BT_SOCKET_L2CAP_CFG_OUT_SENT`, `NG_BT_SOCKET_L2CAP_CLOSED`, `NG_BT_SOCKET_L2CAP_CONFIGURING`, `NG_BT_SOCKET_L2CAP_INFO`, `ng_btsocket_l2cap_pcb_by_cid()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_btsocket_l2cap_untimeout()`, `ng_btsocket_l2cap_pcb::omtu`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::psm`, `ng_btsocket_l2cap_pcb::so`, `ng_btsocket_l2cap_pcb::src`, `ng_btsocket_l2cap_rteentry::src`, and `ng_btsocket_l2cap_pcb::state`.

Referenced by `ng_btsocket_l2cap_l2ca_msg_input()`.

Here is the call graph for this function:



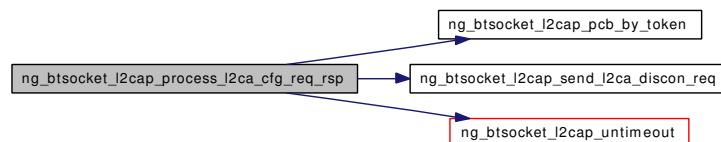
7.67.2.29 static int ng_btsocket_l2cap_process_l2ca_cfg_req_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rtbody_p rt) [static]

Definition at line 663 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cfg_state, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_btsocket_l2cap_pcb::flush_timo, ng_mesg::header, ng_btsocket_l2cap_pcb::imtu, NG_BTsocket_L2CAP_CFG_BOTH, NG_BTsocket_L2CAP_CFG_IN, NG_BTsocket_L2CAP_CFG_IN_SENT, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_CONFIGURING, NG_BTsocket_L2CAP_INFO, NG_BTsocket_L2CAP_OPEN, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_btsocket_l2cap_untimeout(), NG_L2CAP_SUCCESS, NG_L2CAP_UNACCEPTABLE_PARAMS, NG_L2CAP_UNKNOWN_OPTION, ng_btsocket_l2cap_pcb::oflow, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



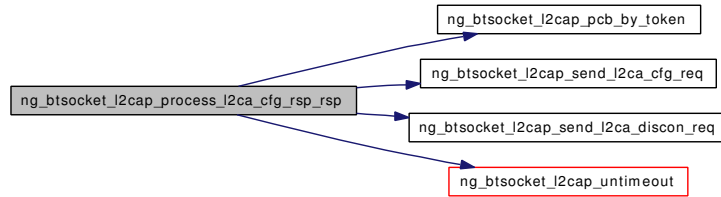
7.67.2.30 static int ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rtbody_p rt) [static]

Definition at line 780 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cfg_state, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_mesg::header, NG_BTsocket_L2CAP_CFG_BOTH, NG_BTsocket_L2CAP_CFG_IN_SENT, NG_BTsocket_L2CAP_CFG_OUT, NG_BTsocket_L2CAP_CFG_OUT_SENT, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_CONFIGURING, NG_BTsocket_L2CAP_INFO, NG_BTsocket_L2CAP_OPEN, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_btsocket_l2cap_untimeout(), NG_L2CAP_SUCCESS, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



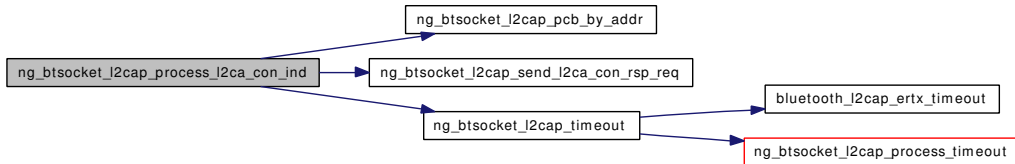
7.67.2.31 static int ng_btsocket_l2cap_process_l2ca_con_ind (struct ng_mesg * msg, ng_btsocket_l2cap_rentry_p rt) [static]

Definition at line 552 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::data, ng_btsocket_l2cap_pcb::flush_timo, ng_mesg::header, ng_btsocket_l2cap_pcb::imtu, NG_BTsocket_L2CAP_CLIENT, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_CONNECTING, NG_BTsocket_L2CAP_INFO, ng_btsocket_l2cap_pcb_by_addr(), ng_btsocket_l2cap_send_l2ca_con_rsp_req(), ng_btsocket_l2cap_untimeout(), NG_HCI_BDADDR_ANY, NG_L2CAP_NO_RESOURCES, NG_L2CAP_PSM_NOT_SUPPORTED, ng_btsocket_l2cap_pcb::oflow, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::so, so2l2cap_pcb, ng_btsocket_l2cap_pcb::src, and ng_btsocket_l2cap_rentry::src.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



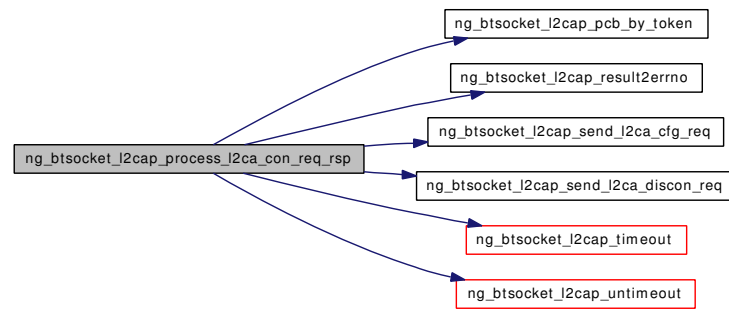
7.67.2.32 static int ng_btsocket_l2cap_process_l2ca_con_req_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rentry_p rt) [static]

Definition at line 388 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cfg_state, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_mesg::header, NG_BTsocket_L2CAP_CFG_IN_SENT, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_CONFIGURING, NG_BTsocket_L2CAP_CONNECTING, NG_BTsocket_L2CAP_INFO, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_result2errno(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_btsocket_l2cap_timeout(), ng_btsocket_l2cap_untimeout(), NG_L2CAP_PENDING, NG_L2CAP_SUCCESS, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



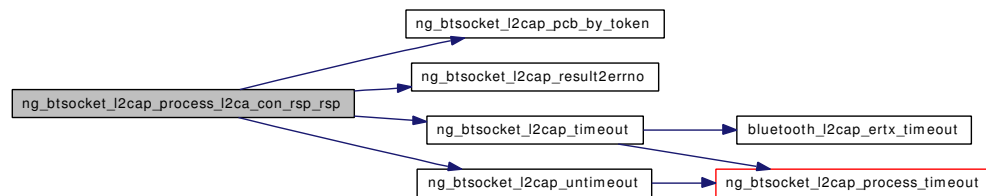
7.67.2.33 static int ng_btsocket_l2cap_process_l2ca_con_rsp_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rtenry_p rt) [static]

Definition at line 485 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cfg_state, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_mesg::header, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_CONFIGURING, NG_BTsocket_L2CAP_CONNECTING, NG_BTsocket_L2CAP_INFO, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_result2errno(), ng_btsocket_l2cap_timeout(), ng_btsocket_l2cap_untimeout(), NG_L2CAP_SUCCESS, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



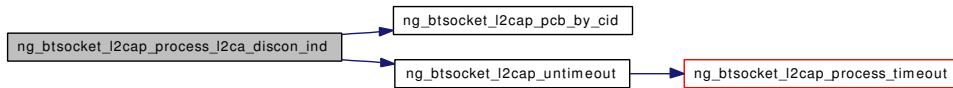
7.67.2.34 static int ng_btsocket_l2cap_process_l2ca_discon_ind (struct ng_mesg * msg, ng_btsocket_l2cap_rtenry_p rt) [static]

Definition at line 1016 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_btsocket_l2cap_pcb::flags, ng_mesg::header, NG_BTsocket_L2CAP_CLOSED, NG_BTsocket_L2CAP_INFO, ng_btsocket_l2cap_pcb_by_cid(), NG_BTsocket_L2CAP_TIMO, ng_btsocket_l2cap_untimeout(), ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_rtenry::src, and ng_btsocket_l2cap_pcb::state.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



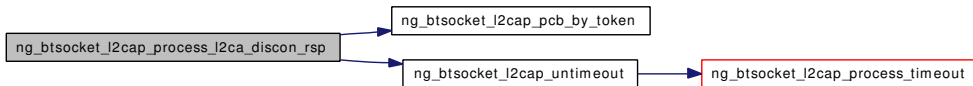
7.67.2.35 static int ng_btsocket_l2cap_process_l2ca_discon_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rteentry_p rt) [static]

Definition at line 959 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_mesg::header, NG_BT_SOCKET_L2CAP_CLOSED, NG_BT_SOCKET_L2CAP_INFO, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_untimeout(), ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



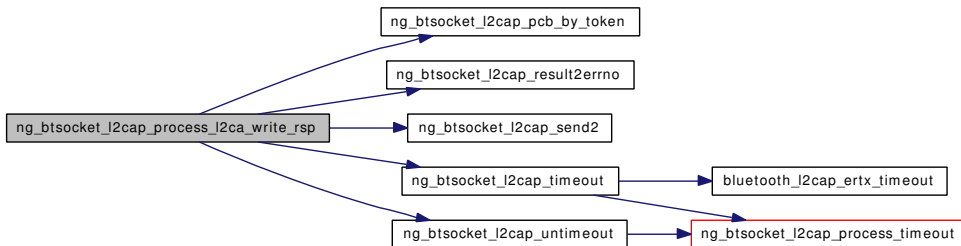
7.67.2.36 static int ng_btsocket_l2cap_process_l2ca_write_rsp (struct ng_mesg * msg, ng_btsocket_l2cap_rteentry_p rt) [static]

Definition at line 1072 of file ng_btsocket_l2cap.c.

References ng_mesg::ng_msghdr::arglen, ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_btsocket_l2cap_pcb::dst, ng_mesg::header, NG_BT_SOCKET_L2CAP_INFO, NG_BT_SOCKET_L2CAP_OPEN, ng_btsocket_l2cap_pcb_by_token(), ng_btsocket_l2cap_result2errno(), ng_btsocket_l2cap_send2(), ng_btsocket_l2cap_timeout(), ng_btsocket_l2cap_untimeout(), ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::so, ng_btsocket_l2cap_pcb::src, ng_btsocket_l2cap_pcb::state, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_l2ca_msg_input().

Here is the call graph for this function:



7.67.2.37 `static void ng_btsocket_l2cap_process_timeout (void * xpcb)` [static]

Definition at line 2678 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::cid`, `ng_btsocket_l2cap_pcb::flags`, `NG_BTSOCKET_L2CAP_CLOSED`, `NG_BTSOCKET_L2CAP_CONFIGURING`, `NG_BTSOCKET_L2CAP_CONNECTING`, `NG_BTSOCKET_L2CAP_DISCONNECTING`, `NG_BTSOCKET_L2CAP_ERR`, `NG_BTSOCKET_L2CAP_OPEN`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `NG_BTSOCKET_L2CAP_TIMO`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::so`, and `ng_btsocket_l2cap_pcb::state`.

Referenced by `ng_btsocket_l2cap_timeout()`, and `ng_btsocket_l2cap_untimeout()`.

Here is the call graph for this function:

**7.67.2.38** `static int ng_btsocket_l2cap_result2errno (int result)` [static]

Definition at line 2727 of file `ng_btsocket_l2cap.c`.

Referenced by `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp()`, and `ng_btsocket_l2cap_process_l2ca_write_rsp()`.

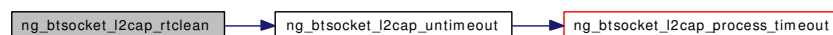
7.67.2.39 `static void ng_btsocket_l2cap_rtclean (void *, int)` [static]

Definition at line 1737 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flags`, `ng_btsocket_l2cap_rtclean::hook`, `NG_BTSOCKET_L2CAP_CLOSED`, `NG_BTSOCKET_L2CAP_TIMO`, `ng_btsocket_l2cap_untimeout()`, `NG_HOOK_NOT_VALID`, `ng_btsocket_l2cap_pcb::pcb_mtx`, and `ng_btsocket_l2cap_pcb::rt`.

Referenced by `ng_btsocket_l2cap_init()`.

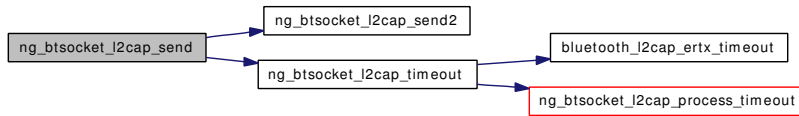
Here is the call graph for this function:

**7.67.2.40** `int ng_btsocket_l2cap_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * nam, struct mbuf * control, struct thread * td)`

Definition at line 2416 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flags`, `ng_btsocket_l2cap_rtclean::hook`, `NG_BTSOCKET_L2CAP_ERR`, `ng_btsocket_l2cap_node`, `NG_BTSOCKET_L2CAP_OPEN`, `ng_btsocket_l2cap_send2()`, `ng_btsocket_l2cap_timeout()`, `NG_BTSOCKET_L2CAP_TIMO`, `NG_FREE_M`, `NG_HOOK_NOT_VALID`, `ng_btsocket_l2cap_pcb::omtu`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::rt`, `ng_btsocket_l2cap_pcb::so`, `so2l2cap_pcb`, and `ng_btsocket_l2cap_pcb::state`.

Here is the call graph for this function:



7.67.2.41 `static int ng_btsocket_l2cap_send2 (ng_btsocket_l2cap_pcb_p pcb)` [static]

Definition at line 2491 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::cid`, `ng_btsocket_l2cap_rtrentry::hook`, `NG_BT_SOCKET_L2CAP_ERR`, `NG_BT_SOCKET_L2CAP_INFO`, `NG_SEND_DATA_ONLY`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::rt`, `ng_btsocket_l2cap_pcb::so`, `ng_btsocket_l2cap_pcb::state`, and `ng_btsocket_l2cap_pcb::token`.

Referenced by `ng_btsocket_l2cap_process_l2ca_write_rsp()`, and `ng_btsocket_l2cap_send()`.

7.67.2.42 `static int ng_btsocket_l2cap_send_l2ca_cfg_req (ng_btsocket_l2cap_pcb_p pcb)` [static]

Definition at line 1214 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::cid`, `ng_mesg::data`, `ng_btsocket_l2cap_pcb::flush_timo`, `ng_mesg::header`, `ng_btsocket_l2cap_rtrentry::hook`, `ng_btsocket_l2cap_pcb::imtu`, `ng_btsocket_l2cap_pcb::link_timo`, `ng_btsocket_l2cap_node`, `NG_HOOK_NOT_VALID`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG`, `ng_btsocket_l2cap_pcb::overflow`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::rt`, `ng_btsocket_l2cap_pcb::token`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, and `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`.

7.67.2.43 `static int ng_btsocket_l2cap_send_l2ca_cfg_rsp (ng_btsocket_l2cap_pcb_p pcb)` [static]

Definition at line 1250 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::cid`, `ng_mesg::data`, `ng_mesg::header`, `ng_btsocket_l2cap_rtrentry::hook`, `ng_btsocket_l2cap_pcb::iflow`, `ng_btsocket_l2cap_node`, `NG_HOOK_NOT_VALID`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CFG_RSP`, `ng_btsocket_l2cap_pcb::omtu`, `ng_btsocket_l2cap_pcb::pcb_mtx`, `ng_btsocket_l2cap_pcb::rt`, `ng_btsocket_l2cap_pcb::token`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_btsocket_l2cap_process_l2ca_cfg_ind()`.

7.67.2.44 `static int ng_btsocket_l2cap_send_l2ca_con_req (ng_btsocket_l2cap_pcb_p pcb)` [static]

Definition at line 1146 of file `ng_btsocket_l2cap.c`.

References `ng_mesg::data`, `ng_btsocket_l2cap_pcb::dst`, `ng_mesg::header`, `ng_btsocket_l2cap_rtrentry::hook`, `ng_btsocket_l2cap_node`, `NG_HOOK_NOT_VALID`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `NGM_L2CAP_L2CA_CON`, `ng_btsocket_l2cap_pcb::pcb_mtx`,

ng_btsocket_l2cap_pcb::psm, ng_btsocket_l2cap_pcb::rt, ng_btsocket_l2cap_pcb::token, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_connect().

7.67.2.45 `static int ng_btsocket_l2cap_send_l2ca_con_rsp_req (u_int32_t token, ng_btsocket_l2cap_rtrentry_p rt, bdaddr_p dst, int ident, int lcid, int result) [static]`

Definition at line 1179 of file ng_btsocket_l2cap.c.

References ng_mesg::data, ng_mesg::header, ng_btsocket_l2cap_rtrentry::hook, ng_btsocket_l2cap_node, NG_HOOK_NOT_VALID, NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_CON_RSP, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_process_l2ca_con_ind().

7.67.2.46 `static int ng_btsocket_l2cap_send_l2ca_discon_req (u_int32_t token, ng_btsocket_l2cap_pcb_p pcb) [static]`

Definition at line 1284 of file ng_btsocket_l2cap.c.

References ng_btsocket_l2cap_pcb::cid, ng_mesg::data, ng_mesg::header, ng_btsocket_l2cap_rtrentry::hook, ng_btsocket_l2cap_node, NG_HOOK_NOT_VALID, NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_DISCON, ng_btsocket_l2cap_pcb::pcb_mtx, ng_btsocket_l2cap_pcb::rt, and ng_mesg::ng_msghdr::token.

Referenced by ng_btsocket_l2cap_detach(), ng_btsocket_l2cap_disconnect(), ng_btsocket_l2cap_process_l2ca_cfg_ind(), ng_btsocket_l2cap_process_l2ca_cfg_req_rsp(), ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), and ng_btsocket_l2cap_process_timeout().

7.67.2.47 `int ng_btsocket_l2cap_sockaddr (struct socket * so, struct sockaddr ** nam)`

Definition at line 2544 of file ng_btsocket_l2cap.c.

References sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_node, ng_btsocket_l2cap_pcb::psm, so2l2cap_pcb, and ng_btsocket_l2cap_pcb::src.

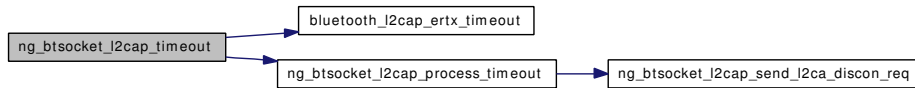
7.67.2.48 `static void ng_btsocket_l2cap_timeout (ng_btsocket_l2cap_pcb_p pcb) [static]`

Definition at line 2643 of file ng_btsocket_l2cap.c.

References bluetooth_l2cap_ertx_timeout(), ng_btsocket_l2cap_pcb::flags, ng_btsocket_l2cap_process_timeout(), NG_BT_SOCKET_L2CAP_TIMO, ng_btsocket_l2cap_pcb::pcb_mtx, and ng_btsocket_l2cap_pcb::timo.

Referenced by ng_btsocket_l2cap_connect(), ng_btsocket_l2cap_disconnect(), ng_btsocket_l2cap_process_l2ca_con_ind(), ng_btsocket_l2cap_process_l2ca_con_req_rsp(), ng_btsocket_l2cap_process_l2ca_con_rsp_rsp(), ng_btsocket_l2cap_process_l2ca_write_rsp(), and ng_btsocket_l2cap_send().

Here is the call graph for this function:



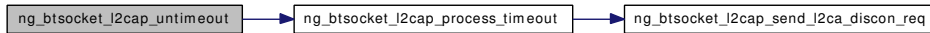
7.67.2.49 static void [ng_btsocket_l2cap_untimeout](#) ([ng_btsocket_l2cap_pcb_pcb](#)) [static]

Definition at line 2661 of file `ng_btsocket_l2cap.c`.

References `ng_btsocket_l2cap_pcb::flags`, `ng_btsocket_l2cap_process_timeout()`, `NG_BT_SOCKET_L2CAP_TIMO`, `ng_btsocket_l2cap_pcb::pcb_mtx`, and `ng_btsocket_l2cap_pcb::timo`.

Referenced by `ng_btsocket_l2cap_detach()`, `ng_btsocket_l2cap_disconnect()`, `ng_btsocket_l2cap_process_l2ca_cfg_ind()`, `ng_btsocket_l2cap_process_l2ca_cfg_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_req_rsp()`, `ng_btsocket_l2cap_process_l2ca_con_rsp_rsp()`, `ng_btsocket_l2cap_process_l2ca_discon_ind()`, `ng_btsocket_l2cap_process_l2ca_discon_rsp()`, `ng_btsocket_l2cap_process_l2ca_write_rsp()`, and `ng_btsocket_l2cap_rtclean()`.

Here is the call graph for this function:



7.67.3 Variable Documentation

7.67.3.1 int [ifqmaxlen](#)

7.67.3.2 u_int32_t [ng_btsocket_l2cap_debug_level](#) [static]

Definition at line 96 of file `ng_btsocket_l2cap.c`.

7.67.3.3 node_p [ng_btsocket_l2cap_node](#) [static]

Definition at line 97 of file `ng_btsocket_l2cap.c`.

Referenced by `ng_btsocket_l2cap_accept()`, `ng_btsocket_l2cap_attach()`, `ng_btsocket_l2cap_bind()`, `ng_btsocket_l2cap_connect()`, `ng_btsocket_l2cap_ctloutput()`, `ng_btsocket_l2cap_detach()`, `ng_btsocket_l2cap_disconnect()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_listen()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_peeraddr()`, `ng_btsocket_l2cap_send()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, and `ng_btsocket_l2cap_sockaddr()`.

7.67.3.4 ng_connect_t [ng_btsocket_l2cap_node_connect](#) [static]

Definition at line 74 of file `ng_btsocket_l2cap.c`.

7.67.3.5 ng_constructor_t [ng_btsocket_l2cap_node_constructor](#) [static]

Definition at line 70 of file `ng_btsocket_l2cap.c`.

7.67.3.6 `ng_disconnect_t ng_btsocket_l2cap_node_disconnect` [static]

Definition at line 76 of file ng_btsocket_l2cap.c.

7.67.3.7 `ng_newhook_t ng_btsocket_l2cap_node_newhook` [static]

Definition at line 73 of file ng_btsocket_l2cap.c.

7.67.3.8 `ng_rcvdata_t ng_btsocket_l2cap_node_rcvdata` [static]

Definition at line 75 of file ng_btsocket_l2cap.c.

7.67.3.9 `ng_rcvmsg_t ng_btsocket_l2cap_node_rcvmsg` [static]

Definition at line 71 of file ng_btsocket_l2cap.c.

7.67.3.10 `ng_shutdown_t ng_btsocket_l2cap_node_shutdown` [static]

Definition at line 72 of file ng_btsocket_l2cap.c.

7.67.3.11 `struct ng_bt_itemq ng_btsocket_l2cap_queue` [static]

Definition at line 98 of file ng_btsocket_l2cap.c.

Referenced by ng_btsocket_l2cap_init(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node_rcvdata(), and ng_btsocket_l2cap_node_rcvmsg().

7.67.3.12 `struct mtx ng_btsocket_l2cap_queue_mtx` [static]

Definition at line 99 of file ng_btsocket_l2cap.c.

Referenced by ng_btsocket_l2cap_init(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node_rcvdata(), and ng_btsocket_l2cap_node_rcvmsg().

7.67.3.13 `struct task ng_btsocket_l2cap_queue_task` [static]

Definition at line 100 of file ng_btsocket_l2cap.c.

Referenced by ng_btsocket_l2cap_init().

7.67.3.14 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_BT_SOCKET_L2CAP_NODE_TYPE,
    .constructor = ng_btsocket_l2cap_node_constructor,
    .rcvmsg =      ng_btsocket_l2cap_node_rcvmsg,
    .shutdown =    ng_btsocket_l2cap_node_shutdown,
    .newhook =     ng_btsocket_l2cap_node_newhook,
```

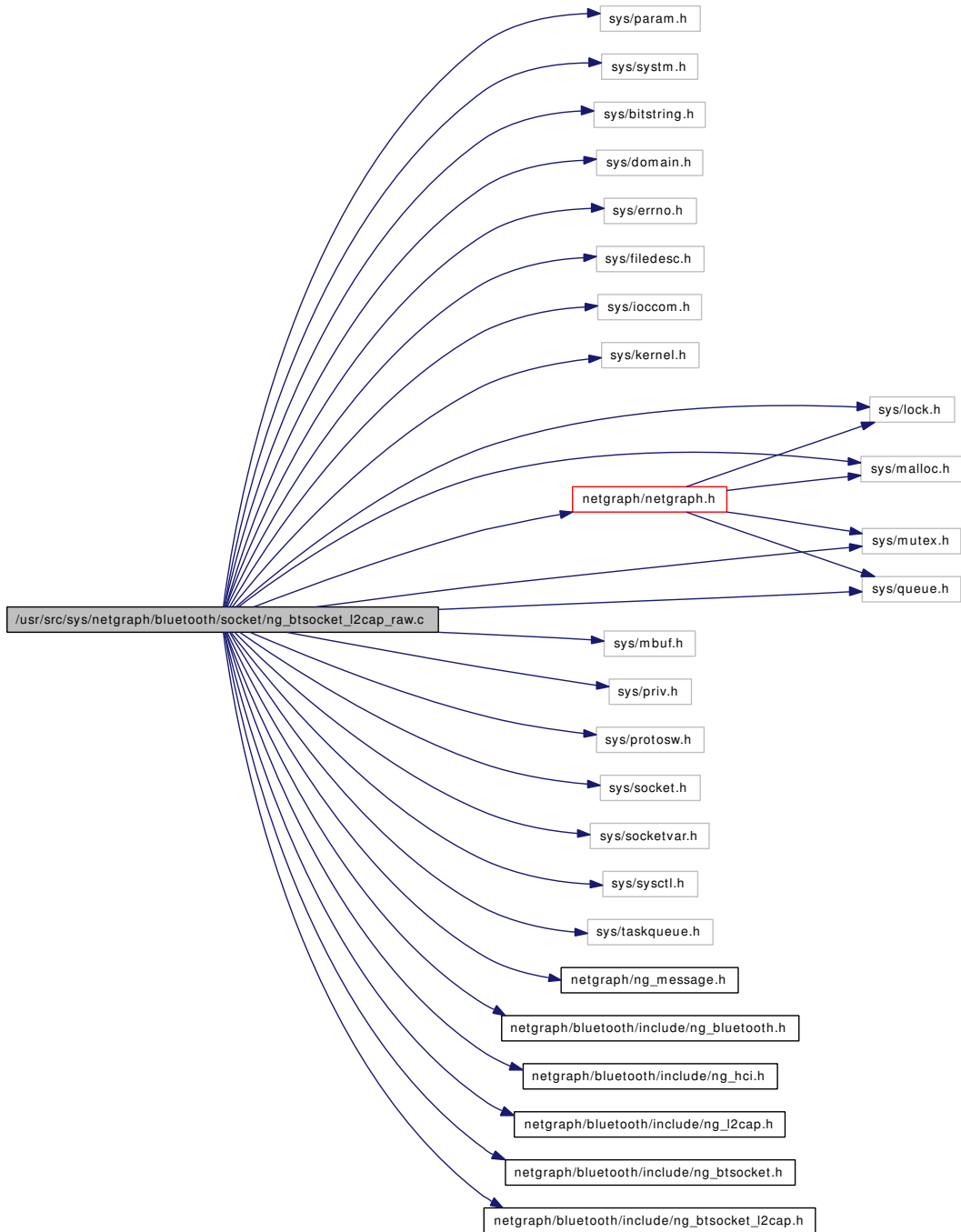
```
.connect =      ng_btsocket_l2cap_node_connect,  
.rcvdata =     ng_btsocket_l2cap_node_rcvdata,  
.disconnect =  ng_btsocket_l2cap_node_disconnect,  
}
```

Definition at line 82 of file ng_btsocket_l2cap.c.

7.68 /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap_raw.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bitstring.h>
#include <sys/domain.h>
#include <sys/errno.h>
#include <sys/filedesc.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/taskqueue.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket.h>
#include <netgraph/bluetooth/include/ng_btsocket_l2cap.h>
```

Include dependency graph for ng_btsocket_l2cap_raw.c:



Defines

- `#define M_NETGRAPH_BT SOCKET_L2CAP_RAW M_NETGRAPH`
- `#define ng_btsocket_l2cap_raw_wakeup_input_task() taskqueue_enqueue(taskqueue_swi, &ng_btsocket_l2cap_raw_queue_task)`
- `#define ng_btsocket_l2cap_raw_wakeup_route_task() taskqueue_enqueue(taskqueue_swi, &ng_btsocket_l2cap_raw_rt_task)`
- `#define NG_BT SOCKET_L2CAP_RAW_INFO`

- #define [NG_BT_SOCKET_L2CAP_RAW_WARN](#)
- #define [NG_BT_SOCKET_L2CAP_RAW_ERR](#)
- #define [NG_BT_SOCKET_L2CAP_RAW_ALERT](#)

Functions

- static void [ng_btsocket_l2cap_raw_input](#) (void *, int)
- static void [ng_btsocket_l2cap_raw_rtclean](#) (void *, int)
- static void [ng_btsocket_l2cap_raw_get_token](#) (u_int32_t *)
- static int [ng_btsocket_l2cap_raw_send_ngmsg](#) (hook_p, int, void *, int)
- static int [ng_btsocket_l2cap_raw_send_sync_ngmsg](#) (ng_btsocket_l2cap_raw_pcb_p, int, void *, int)
- static LIST_HEAD ([ng_btsocket_l2cap_raw_pcb](#))
- static int [ng_btsocket_l2cap_raw_node_shutdown](#) (node_p node)
- static int [ng_btsocket_l2cap_raw_node_newhook](#) (node_p node, hook_p hook, char const *name)
- static int [ng_btsocket_l2cap_raw_node_connect](#) (hook_p hook)
- static int [ng_btsocket_l2cap_raw_node_disconnect](#) (hook_p hook)
- static int [ng_btsocket_l2cap_raw_node_rcvmsg](#) (node_p node, item_p item, hook_p hook)
- static int [ng_btsocket_l2cap_raw_node_rcvdata](#) (hook_p hook, item_p item)
- void [ng_btsocket_l2cap_raw_init](#) (void)
- void [ng_btsocket_l2cap_raw_abort](#) (struct socket *so)
- void [ng_btsocket_l2cap_raw_close](#) (struct socket *so)
- int [ng_btsocket_l2cap_raw_attach](#) (struct socket *so, int proto, struct thread *td)
- int [ng_btsocket_l2cap_raw_bind](#) (struct socket *so, struct sockaddr *nam, struct thread *td)
- int [ng_btsocket_l2cap_raw_connect](#) (struct socket *so, struct sockaddr *nam, struct thread *td)
- int [ng_btsocket_l2cap_raw_control](#) (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- void [ng_btsocket_l2cap_raw_detach](#) (struct socket *so)
- int [ng_btsocket_l2cap_raw_disconnect](#) (struct socket *so)
- int [ng_btsocket_l2cap_raw_peeraddr](#) (struct socket *so, struct sockaddr **nam)
- int [ng_btsocket_l2cap_raw_send](#) (struct socket *so, int flags, struct mbuf *m, struct sockaddr *nam, struct mbuf *control, struct thread *td)
- int [ng_btsocket_l2cap_raw_sockaddr](#) (struct socket *so, struct sockaddr **nam)

Variables

- static [ng_constructor_t](#) [ng_btsocket_l2cap_raw_node_constructor](#)
- static [ng_rcvmsg_t](#) [ng_btsocket_l2cap_raw_node_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_btsocket_l2cap_raw_node_shutdown](#)
- static [ng_newhook_t](#) [ng_btsocket_l2cap_raw_node_newhook](#)
- static [ng_connect_t](#) [ng_btsocket_l2cap_raw_node_connect](#)
- static [ng_rcvdata_t](#) [ng_btsocket_l2cap_raw_node_rcvdata](#)
- static [ng_disconnect_t](#) [ng_btsocket_l2cap_raw_node_disconnect](#)
- static struct [ng_type](#) typestruct
- int ifqmaxlen
- static u_int32_t [ng_btsocket_l2cap_raw_debug_level](#)
- static u_int32_t [ng_btsocket_l2cap_raw_ioctl_timeout](#)
- static [node_p](#) [ng_btsocket_l2cap_raw_node](#)
- static struct [ng_bt_itemq](#) [ng_btsocket_l2cap_raw_queue](#)
- static struct mtx [ng_btsocket_l2cap_raw_queue_mtx](#)
- static struct task [ng_btsocket_l2cap_raw_queue_task](#)

7.68.1 Define Documentation

7.68.1.1 #define M_NETGRAPH_BT SOCKET_L2CAP_RAW M_NETGRAPH

Definition at line 66 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_attach(), ng_btsocket_l2cap_raw_detach(), ng_btsocket_l2cap_raw_input(), and ng_btsocket_l2cap_raw_rtclean().

7.68.1.2 #define NG_BT SOCKET_L2CAP_RAW_ALERT

Value:

```
if (ng_btsocket_l2cap_raw_debug_level >= NG_BT SOCKET_ALERT_LEVEL) \  
    printf
```

Referenced by ng_btsocket_l2cap_raw_init(), and ng_btsocket_l2cap_raw_node_shutdown().

7.68.1.3 #define NG_BT SOCKET_L2CAP_RAW_ERR

Value:

```
if (ng_btsocket_l2cap_raw_debug_level >= NG_BT SOCKET_ERR_LEVEL) \  
    printf
```

Referenced by ng_btsocket_l2cap_raw_node_rcvmsg().

7.68.1.4 #define NG_BT SOCKET_L2CAP_RAW_INFO

Value:

```
if (ng_btsocket_l2cap_raw_debug_level >= NG_BT SOCKET_INFO_LEVEL) \  
    printf
```

Referenced by ng_btsocket_l2cap_raw_input().

7.68.1.5 #define ng_btsocket_l2cap_raw_wakeup_input_task() taskqueue_enqueue(taskqueue_ - swi, &ng_btsocket_l2cap_raw_queue_task)

Definition at line 87 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_node_rcvmsg().

7.68.1.6 #define ng_btsocket_l2cap_raw_wakeup_route_task() taskqueue_enqueue(taskqueue_ - swi, &ng_btsocket_l2cap_raw_rt_task)

Definition at line 90 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_node_disconnect().

7.68.1.7 #define NG_BT_SOCKET_L2CAP_RAW_WARN

Value:

```
if (ng_btsocket_l2cap_raw_debug_level >= NG_BT_SOCKET_WARN_LEVEL) \
    printf
```

Referenced by ng_btsocket_l2cap_raw_input().

7.68.2 Function Documentation

7.68.2.1 static LIST_HEAD (ng_btsocket_l2cap_raw_pcb) [static]

Definition at line 114 of file ng_btsocket_l2cap_raw.c.

7.68.2.2 void ng_btsocket_l2cap_raw_abort (struct socket * so)

Definition at line 577 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_disconnect().

Here is the call graph for this function:



7.68.2.3 int ng_btsocket_l2cap_raw_attach (struct socket * so, int proto, struct thread * td)

Definition at line 595 of file ng_btsocket_l2cap_raw.c.

References M_NETGRAPH_BT_SOCKET_L2CAP_RAW, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_PRIVILEGED, NG_BT_SOCKET_L2CAP_RAW_RECVSPACE, NG_BT_SOCKET_L2CAP_RAW_SENDSPACE, and so2l2cap_raw_pcb.

7.68.2.4 int ng_btsocket_l2cap_raw_bind (struct socket * so, struct sockaddr * nam, struct thread * td)

Definition at line 642 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_rtentry::hook, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, ng_btsocket_l2cap_raw_node, NG_HCI_BDADDR_ANY, NG_HOOK_NOT_VALID, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, so2l2cap_raw_pcb, ng_btsocket_l2cap_raw_pcb::src, and ng_btsocket_l2cap_rtentry::src.

7.68.2.5 void ng_btsocket_l2cap_raw_close (struct socket * so)

Definition at line 584 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_disconnect().

Here is the call graph for this function:



7.68.2.6 int ng_btsocket_l2cap_raw_connect (struct socket * so, struct sockaddr * nam, struct thread * td)

Definition at line 694 of file ng_btsocket_l2cap_raw.c.

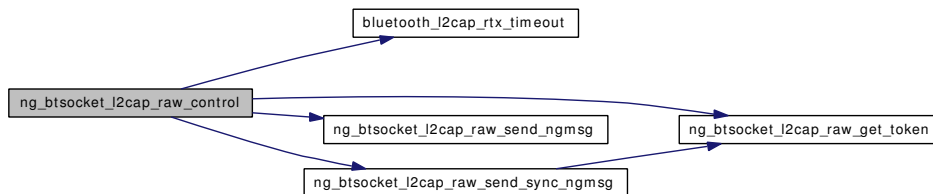
References ng_btsocket_l2cap_raw_pcb::dst, ng_btsocket_l2cap_rtrentry::hook, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, ng_btsocket_l2cap_raw_node, NG_HCI_BDADDR_ANY, NG_HOOK_NOT_VALID, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, so2l2cap_raw_pcb, ng_btsocket_l2cap_rtrentry::src, and ng_btsocket_l2cap_raw_pcb::src.

7.68.2.7 int ng_btsocket_l2cap_raw_control (struct socket * so, u_long cmd, caddr_t data, struct ifnet * ifp, struct thread * td)

Definition at line 772 of file ng_btsocket_l2cap_raw.c.

References bluetooth_l2cap_rtx_timeout(), ng_btsocket_l2cap_raw_chan_list::channels, ng_mesg::ng_msghdr::cmd, ng_btsocket_l2cap_raw_con_list::connections, ng_mesg::data, ng_btsocket_l2cap_raw_node_debug::debug, ng_btsocket_l2cap_raw_pcb::dst, ng_btsocket_l2cap_raw_ping::echo_data, ng_l2cap_l2ca_ping_op::echo_size, ng_btsocket_l2cap_raw_ping::echo_size, ng_btsocket_l2cap_raw_pcb::flags, ng_btsocket_l2cap_raw_node_flags::flags, ng_mesg::header, ng_btsocket_l2cap_rtrentry::hook, ng_btsocket_l2cap_raw_get_info::info_data, ng_l2cap_l2ca_get_info_op::info_size, ng_btsocket_l2cap_raw_get_info::info_size, ng_btsocket_l2cap_raw_get_info::info_type, min, ng_btsocket_l2cap_raw_pcb::msg, ng_btsocket_l2cap_raw_get_token(), ng_btsocket_l2cap_raw_ioctl_timeout, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_PRIVILEGED, ng_btsocket_l2cap_raw_send_ngmsg(), ng_btsocket_l2cap_raw_send_sync_ngmsg(), NG_FREE_MSG, NG_L2CAP_MAX_CHAN_NUM, NG_L2CAP_MAX_CON_NUM, NG_L2CAP_MAX_ECHO_SIZE, NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_L2CA_GET_INFO, NGM_L2CAP_L2CA_PING, NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO, NGM_L2CAP_NODE_GET_CHAN_LIST, NGM_L2CAP_NODE_GET_CON_LIST, NGM_L2CAP_NODE_GET_DEBUG, NGM_L2CAP_NODE_GET_FLAGS, NGM_L2CAP_NODE_SET_AUTO_DISCON_TIMO, NGM_L2CAP_NODE_SET_DEBUG, ng_l2cap_node_chan_list_ep::num_channels, ng_btsocket_l2cap_raw_chan_list::num_channels, ng_l2cap_node_con_list_ep::num_connections, ng_btsocket_l2cap_raw_con_list::num_connections, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_l2cap_l2ca_get_info_op::result, ng_btsocket_l2cap_raw_get_info::result, ng_l2cap_l2ca_ping_op::result, ng_btsocket_l2cap_raw_ping::result, ng_btsocket_l2cap_raw_pcb::rt, SIOC_L2CAP_L2CA_GET_INFO, SIOC_L2CAP_L2CA_PING, SIOC_L2CAP_NODE_GET_AUTO_DISCON_TIMO, SIOC_L2CAP_NODE_GET_CHAN_LIST, SIOC_L2CAP_NODE_GET_CON_LIST, SIOC_L2CAP_NODE_GET_DEBUG, SIOC_L2CAP_NODE_GET_FLAGS, SIOC_L2CAP_NODE_SET_AUTO_DISCON_TIMO, SIOC_L2CAP_NODE_SET_DEBUG, so2l2cap_raw_pcb, ng_btsocket_l2cap_raw_auto_discon_timo::timeout, ng_mesg::ng_msghdr::token, and ng_btsocket_l2cap_raw_pcb::token.

Here is the call graph for this function:



7.68.2.8 void ng_btsocket_l2cap_raw_detach (struct socket * so)

Definition at line 1107 of file ng_btsocket_l2cap_raw.c.

References M_NETGRAPH_BT_SOCKET_L2CAP_RAW, ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, and so2l2cap_raw_pcb.

7.68.2.9 int ng_btsocket_l2cap_raw_disconnect (struct socket * so)

Definition at line 1136 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, and so2l2cap_raw_pcb.

Referenced by ng_btsocket_l2cap_raw_abort(), and ng_btsocket_l2cap_raw_close().

7.68.2.10 static void ng_btsocket_l2cap_raw_get_token (u_int32_t *) [static]

Definition at line 1228 of file ng_btsocket_l2cap_raw.c.

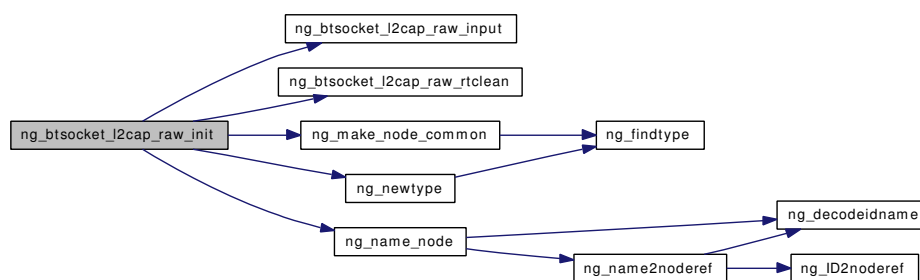
Referenced by ng_btsocket_l2cap_raw_control(), and ng_btsocket_l2cap_raw_send_sync_ngmsg().

7.68.2.11 void ng_btsocket_l2cap_raw_init (void)

Definition at line 507 of file ng_btsocket_l2cap_raw.c.

References ifqmaxlen, NG_BT_ITEMQ_INIT, NG_BT_SOCKET_L2CAP_RAW_ALERT, ng_btsocket_l2cap_raw_debug_level, ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_ioctl_timeout, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE, ng_btsocket_l2cap_raw_queue, ng_btsocket_l2cap_raw_queue_mtx, ng_btsocket_l2cap_raw_queue_task, ng_btsocket_l2cap_raw_rtclean(), NG_BT_SOCKET_WARN_LEVEL, ng_make_node_common(), ng_name_node(), ng_newtype(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:

**7.68.2.12 static void ng_btsocket_l2cap_raw_input (void *, int) [static]**

Definition at line 338 of file ng_btsocket_l2cap_raw.c.

References ng_msg::ng_msghdr::arglen, ng_msg::ng_msghdr::cmd, ng_msg::data, ng_item::el_flags, ng_msg::header, ng_btsocket_l2cap_rtentry::hook, M_NETGRAPH_BT_SOCKET_L2CAP_RAW, ng_btsocket_l2cap_raw_pcb::msg, NG_BT_ITEMQ_DEQUEUE, NG_BT_SOCKET_L2CAP_RAW_INFO,

ng_btsocket_l2cap_raw_queue, ng_btsocket_l2cap_raw_queue_mtx, NG_BT_SOCKET_L2CAP_RAW_WARN, NG_FREE_ITEM, NG_FREE_MSG, NG_HCI_BDADDR_ANY, NG_HOOK_NAME, NG_HOOK_NOT_VALID, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_HOOK_UNREF, NGI_GET_HOOK, NGI_GET_MSG, NGM_L2CAP_L2CA_GET_INFO, NGM_L2CAP_L2CA_PING, NGM_L2CAP_NODE_GET_AUTO_DISCON_TIMO, NGM_L2CAP_NODE_GET_CHAN_LIST, NGM_L2CAP_NODE_GET_CON_LIST, NGM_L2CAP_NODE_GET_DEBUG, NGM_L2CAP_NODE_GET_FLAGS, NGM_L2CAP_NODE_HOOK_INFO, NGQF_MESG, NGQF_TYPE, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_rtrentry::src, ng_mesg::ng_msghdr::token, and ng_btsocket_l2cap_raw_pcb::token.

Referenced by ng_btsocket_l2cap_raw_init().

7.68.2.13 static int ng_btsocket_l2cap_raw_node_connect ([hook_p hook](#)) [static]

Definition at line 232 of file ng_btsocket_l2cap_raw.c.

References NG_HOOK_REF, and NG_HOOK_SET_PRIVATE.

7.68.2.14 static int ng_btsocket_l2cap_raw_node_disconnect ([hook_p hook](#)) [static]

Definition at line 245 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_wakeup_route_task, NG_HOOK_PRIVATE, and NG_HOOK_UNREF.

7.68.2.15 static int ng_btsocket_l2cap_raw_node_newhook ([node_p node](#), [hook_p hook](#), char const * [name](#)) [static]

Definition at line 222 of file ng_btsocket_l2cap_raw.c.

7.68.2.16 static int ng_btsocket_l2cap_raw_node_rcvdata ([hook_p hook](#), [item_p item](#)) [static]

Definition at line 320 of file ng_btsocket_l2cap_raw.c.

References NG_FREE_ITEM.

7.68.2.17 static int ng_btsocket_l2cap_raw_node_rcvmsg ([node_p node](#), [item_p item](#), [hook_p hook](#)) [static]

Definition at line 268 of file ng_btsocket_l2cap_raw.c.

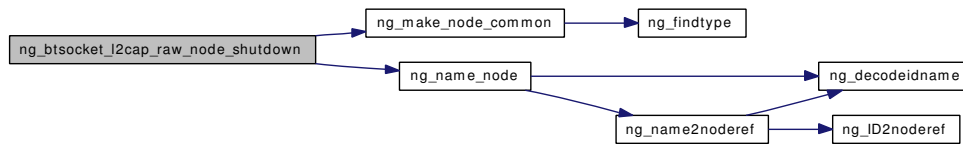
References ng_mesg::ng_msghdr::cmd, ng_mesg::ng_msghdr::flags, ng_mesg::header, NG_BT_ITEMQ_DROP, NG_BT_ITEMQ_ENQUEUE, NG_BT_ITEMQ_FULL, NG_BT_SOCKET_L2CAP_RAW_ERR, ng_btsocket_l2cap_raw_queue, ng_btsocket_l2cap_raw_queue_mtx, ng_btsocket_l2cap_raw_wakeup_input_task, NG_FREE_ITEM, NG_HOOK_REF, NGF_RESP, NGI_MSG, NGI_SET_HOOK, NGM_L2CAP_COOKIE, NGM_L2CAP_NODE_HOOK_INFO, ng_mesg::ng_msghdr::token, and ng_mesg::ng_msghdr::typecookie.

7.68.2.18 static int ng_btsocket_l2cap_raw_node_shutdown ([node_p node](#)) [static]

Definition at line 185 of file ng_btsocket_l2cap_raw.c.

References NG_BT_SOCKET_L2CAP_RAW_ALERT, ng_btsocket_l2cap_raw_node, NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE, ng_make_node_common(), ng_name_node(), NG_NODE_UNREF, and typestruct.

Here is the call graph for this function:



7.68.2.19 int ng_btsocket_l2cap_raw_peeraddr (struct socket * so, struct sockaddr ** nam)

Definition at line 1158 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_raw_pcb::dst, sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, ng_btsocket_l2cap_raw_node, ng_btsocket_l2cap_raw_pcb::pcb_mtx, and so2l2cap_raw_pcb.

7.68.2.20 static void ng_btsocket_l2cap_raw_rtclean (void *, int) [static]

Definition at line 449 of file ng_btsocket_l2cap_raw.c.

References ng_btsocket_l2cap_rtentry::hook, M_NETGRAPH_BT_SOCKET_L2CAP_RAW, NG_HOOK_NOT_VALID, NG_HOOK_SET_PRIVATE, NG_HOOK_UNREF, ng_btsocket_l2cap_raw_pcb::pcb_mtx, ng_btsocket_l2cap_raw_pcb::rt, and ng_btsocket_l2cap_raw_pcb::so.

Referenced by ng_btsocket_l2cap_raw_init().

7.68.2.21 int ng_btsocket_l2cap_raw_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * nam, struct mbuf * control, struct thread * td)

Definition at line 1186 of file ng_btsocket_l2cap_raw.c.

References NG_FREE_M.

7.68.2.22 static int ng_btsocket_l2cap_raw_send_ngmsg (hook_p, int, void *, int) [static]

Definition at line 1245 of file ng_btsocket_l2cap_raw.c.

References ng_mesg::data, ng_btsocket_l2cap_raw_node, NG_MKMESSAGE, NG_SEND_MSG_HOOK, and NGM_L2CAP_COOKIE.

Referenced by ng_btsocket_l2cap_raw_control().

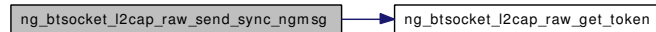
7.68.2.23 static int ng_btsocket_l2cap_raw_send_sync_ngmsg (ng_btsocket_l2cap_raw_pcb_p, int, void *, int) [static]

Definition at line 1267 of file ng_btsocket_l2cap_raw.c.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `ng_btsocket_l2cap_rtrentry::hook`, `ng_btsocket_l2cap_raw_pcb::msg`, `ng_btsocket_l2cap_raw_get_token()`, `ng_btsocket_l2cap_raw_ioctl_timeout`, `ng_btsocket_l2cap_raw_node`, `NG_FREE_MSG`, `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, `NGM_L2CAP_COOKIE`, `ng_btsocket_l2cap_raw_pcb::pcb_mtx`, `ng_btsocket_l2cap_raw_pcb::rt`, `ng_btsocket_l2cap_raw_pcb::token`, and `ng_mesg::ng_msghdr::token`.

Referenced by `ng_btsocket_l2cap_raw_control()`.

Here is the call graph for this function:



7.68.2.24 `int ng_btsocket_l2cap_raw_sockaddr (struct socket * so, struct sockaddr ** nam)`

Definition at line 1200 of file `ng_btsocket_l2cap_raw.c`.

References `sockaddr_l2cap::l2cap_bdaddr`, `sockaddr_l2cap::l2cap_family`, `sockaddr_l2cap::l2cap_len`, `sockaddr_l2cap::l2cap_psm`, `ng_btsocket_l2cap_raw_node`, `ng_btsocket_l2cap_raw_pcb::pcb_mtx`, `so2l2cap_raw_pcb`, and `ng_btsocket_l2cap_raw_pcb::src`.

7.68.3 Variable Documentation

7.68.3.1 `int ifqmaxlen`

7.68.3.2 `u_int32_t ng_btsocket_l2cap_raw_debug_level` [static]

Definition at line 108 of file `ng_btsocket_l2cap_raw.c`.

Referenced by `ng_btsocket_l2cap_raw_init()`.

7.68.3.3 `u_int32_t ng_btsocket_l2cap_raw_ioctl_timeout` [static]

Definition at line 109 of file `ng_btsocket_l2cap_raw.c`.

Referenced by `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_init()`, and `ng_btsocket_l2cap_raw_send_sync_ngmsg()`.

7.68.3.4 `node_p ng_btsocket_l2cap_raw_node` [static]

Definition at line 110 of file `ng_btsocket_l2cap_raw.c`.

Referenced by `ng_btsocket_l2cap_raw_attach()`, `ng_btsocket_l2cap_raw_bind()`, `ng_btsocket_l2cap_raw_connect()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_detach()`, `ng_btsocket_l2cap_raw_disconnect()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_btsocket_l2cap_raw_peeraddr()`, `ng_btsocket_l2cap_raw_send_ngmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, and `ng_btsocket_l2cap_raw_sockaddr()`.

7.68.3.5 `ng_connect_t ng_btsocket_l2cap_raw_node_connect` [static]

Definition at line 74 of file `ng_btsocket_l2cap_raw.c`.

7.68.3.6 [ng_constructor_t ng_btsocket_l2cap_raw_node_constructor](#) [static]

Definition at line 70 of file ng_btsocket_l2cap_raw.c.

7.68.3.7 [ng_disconnect_t ng_btsocket_l2cap_raw_node_disconnect](#) [static]

Definition at line 76 of file ng_btsocket_l2cap_raw.c.

7.68.3.8 [ng_newhook_t ng_btsocket_l2cap_raw_node_newhook](#) [static]

Definition at line 73 of file ng_btsocket_l2cap_raw.c.

7.68.3.9 [ng_rcvdata_t ng_btsocket_l2cap_raw_node_rcvdata](#) [static]

Definition at line 75 of file ng_btsocket_l2cap_raw.c.

7.68.3.10 [ng_rcvmsg_t ng_btsocket_l2cap_raw_node_rcvmsg](#) [static]

Definition at line 71 of file ng_btsocket_l2cap_raw.c.

7.68.3.11 [ng_shutdown_t ng_btsocket_l2cap_raw_node_shutdown](#) [static]

Definition at line 72 of file ng_btsocket_l2cap_raw.c.

7.68.3.12 **struct** [ng_bt_itemq ng_btsocket_l2cap_raw_queue](#) [static]

Definition at line 111 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_init(), ng_btsocket_l2cap_raw_input(), and ng_btsocket_l2cap_raw_node_rcvmsg().

7.68.3.13 **struct** **mtx** [ng_btsocket_l2cap_raw_queue_mtx](#) [static]

Definition at line 112 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_init(), ng_btsocket_l2cap_raw_input(), and ng_btsocket_l2cap_raw_node_rcvmsg().

7.68.3.14 **struct** **task** [ng_btsocket_l2cap_raw_queue_task](#) [static]

Definition at line 113 of file ng_btsocket_l2cap_raw.c.

Referenced by ng_btsocket_l2cap_raw_init().

7.68.3.15 **struct** [ng_type_t](#) **typestruct** [static]

Initial value:

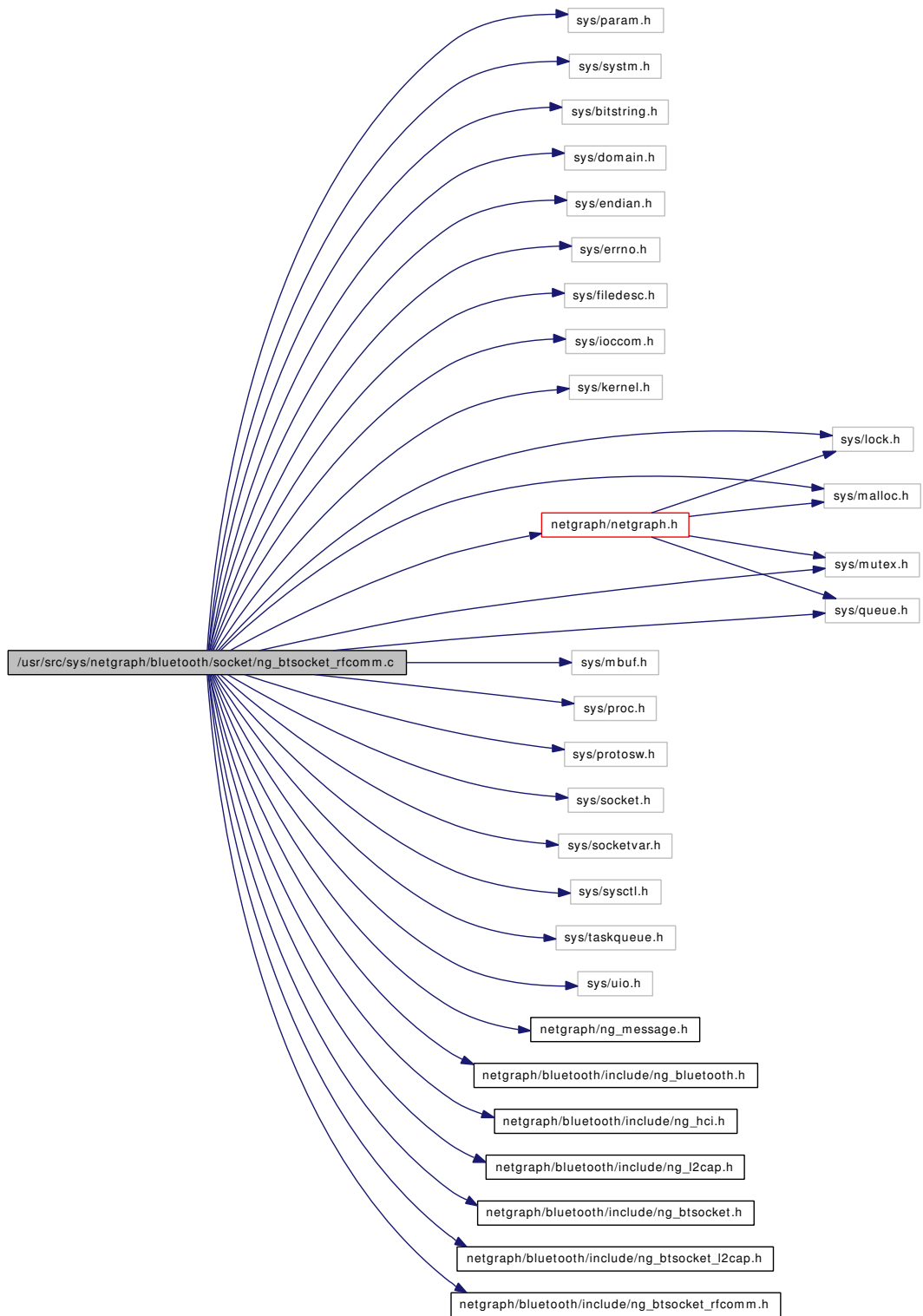
```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE,
    .constructor = ng_btsocket_l2cap_raw_node_constructor,
    .rcvmsg =      ng_btsocket_l2cap_raw_node_rcvmsg,
    .shutdown =    ng_btsocket_l2cap_raw_node_shutdown,
    .newhook =     ng_btsocket_l2cap_raw_node_newhook,
    .connect =     ng_btsocket_l2cap_raw_node_connect,
    .rcvdata =     ng_btsocket_l2cap_raw_node_rcvdata,
    .disconnect =  ng_btsocket_l2cap_raw_node_disconnect,
}
```

Definition at line 94 of file `ng_btsocket_l2cap_raw.c`.

7.69 /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_rfcomm.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/bitstring.h>
#include <sys/domain.h>
#include <sys/endian.h>
#include <sys/errno.h>
#include <sys/filedesc.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/proc.h>
#include <sys/protosw.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/taskqueue.h>
#include <sys/uio.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/bluetooth/include/ng_bluetooth.h>
#include <netgraph/bluetooth/include/ng_hci.h>
#include <netgraph/bluetooth/include/ng_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket.h>
#include <netgraph/bluetooth/include/ng_btsocket_l2cap.h>
#include <netgraph/bluetooth/include/ng_btsocket_rfcomm.h>
```

Include dependency graph for ng_btsocket_rfcomm.c:



Defines

- #define [M_NETGRAPH_BT_SOCKET_RFCOMM](#) M_NETGRAPH

- #define NG_BT_SOCKET_RFCOMM_INFO
- #define NG_BT_SOCKET_RFCOMM_WARN
- #define NG_BT_SOCKET_RFCOMM_ERR
- #define NG_BT_SOCKET_RFCOMM_ALERT
- #define ALOT 0x7fff
- #define ng_btsocket_rfcomm_task_wakeup() taskqueue_enqueue(taskqueue_swi_giant, &ng_btsocket_rfcomm_task)

Functions

- static void ng_btsocket_rfcomm_upcall (struct socket *so, void *arg, int waitflag)
- static void ng_btsocket_rfcomm_sessions_task (void *ctx, int pending)
- static void ng_btsocket_rfcomm_session_task (ng_btsocket_rfcomm_session_p s)
- static ng_btsocket_rfcomm_pcb_p ng_btsocket_rfcomm_connect_ind (ng_btsocket_rfcomm_session_p s, int channel)
- static void ng_btsocket_rfcomm_connect_cfm (ng_btsocket_rfcomm_session_p s)
- static int ng_btsocket_rfcomm_session_create (ng_btsocket_rfcomm_session_p *sp, struct socket *l2so, bdaddr_p src, bdaddr_p dst, struct thread *td)
- static int ng_btsocket_rfcomm_session_accept (ng_btsocket_rfcomm_session_p s0)
- static int ng_btsocket_rfcomm_session_connect (ng_btsocket_rfcomm_session_p s)
- static int ng_btsocket_rfcomm_session_receive (ng_btsocket_rfcomm_session_p s)
- static int ng_btsocket_rfcomm_session_send (ng_btsocket_rfcomm_session_p s)
- static void ng_btsocket_rfcomm_session_clean (ng_btsocket_rfcomm_session_p s)
- static void ng_btsocket_rfcomm_session_process_pcb (ng_btsocket_rfcomm_session_p s)
- static ng_btsocket_rfcomm_session_p ng_btsocket_rfcomm_session_by_addr (bdaddr_p src, bdaddr_p dst)
- static int ng_btsocket_rfcomm_receive_frame (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_sabm (ng_btsocket_rfcomm_session_p s, int dlc_i)
- static int ng_btsocket_rfcomm_receive_disc (ng_btsocket_rfcomm_session_p s, int dlc_i)
- static int ng_btsocket_rfcomm_receive_ua (ng_btsocket_rfcomm_session_p s, int dlc_i)
- static int ng_btsocket_rfcomm_receive_dm (ng_btsocket_rfcomm_session_p s, int dlc_i)
- static int ng_btsocket_rfcomm_receive_uih (ng_btsocket_rfcomm_session_p s, int dlc_i, int pf, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_mcc (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_test (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_fc (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_msc (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_rpn (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_rls (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static int ng_btsocket_rfcomm_receive_pn (ng_btsocket_rfcomm_session_p s, struct mbuf *m0)
- static void ng_btsocket_rfcomm_set_pn (ng_btsocket_rfcomm_pcb_p pcb, u_int8_t cr, u_int8_t flow_control, u_int8_t credits, u_int16_t mtu)
- static int ng_btsocket_rfcomm_send_command (ng_btsocket_rfcomm_session_p s, u_int8_t type, u_int8_t dlc_i)
- static int ng_btsocket_rfcomm_send_uih (ng_btsocket_rfcomm_session_p s, u_int8_t address, u_int8_t pf, u_int8_t credits, struct mbuf *data)
- static int ng_btsocket_rfcomm_send_msc (ng_btsocket_rfcomm_pcb_p pcb)
- static int ng_btsocket_rfcomm_send_pn (ng_btsocket_rfcomm_pcb_p pcb)
- static int ng_btsocket_rfcomm_send_credits (ng_btsocket_rfcomm_pcb_p pcb)
- static int ng_btsocket_rfcomm_pcb_send (ng_btsocket_rfcomm_pcb_p pcb, int limit)

- static void `ng_btsocket_rfcomm_pcb_kill` (`ng_btsocket_rfcomm_pcb_p` pcb, int error)
- static `ng_btsocket_rfcomm_pcb_p` `ng_btsocket_rfcomm_pcb_by_channel` (`bdaddr_p` src, int channel)
- static `ng_btsocket_rfcomm_pcb_p` `ng_btsocket_rfcomm_pcb_by_dlc` (`ng_btsocket_rfcomm_session_p` s, int dlc)
- static `ng_btsocket_rfcomm_pcb_p` `ng_btsocket_rfcomm_pcb_listener` (`bdaddr_p` src, int channel)
- static void `ng_btsocket_rfcomm_timeout` (`ng_btsocket_rfcomm_pcb_p` pcb)
- static void `ng_btsocket_rfcomm_untimeout` (`ng_btsocket_rfcomm_pcb_p` pcb)
- static void `ng_btsocket_rfcomm_process_timeout` (void *xpcb)
- static struct mbuf * `ng_btsocket_rfcomm_prepare_packet` (struct sockbuf *sb, int length)
- static LIST_HEAD (`ng_btsocket_rfcomm_session`)
- static u_int8_t `ng_btsocket_rfcomm_crc` (u_int8_t *data, int length)
- static u_int8_t `ng_btsocket_rfcomm_fcs2` (u_int8_t *data)
- static u_int8_t `ng_btsocket_rfcomm_fcs3` (u_int8_t *data)
- static int `ng_btsocket_rfcomm_check_fcs` (u_int8_t *data, int type, u_int8_t fcs)
- void `ng_btsocket_rfcomm_init` (void)
- void `ng_btsocket_rfcomm_abort` (struct socket *so)
- void `ng_btsocket_rfcomm_close` (struct socket *so)
- int `ng_btsocket_rfcomm_accept` (struct socket *so, struct sockaddr **nam)
- int `ng_btsocket_rfcomm_attach` (struct socket *so, int proto, struct thread *td)
- int `ng_btsocket_rfcomm_bind` (struct socket *so, struct sockaddr *nam, struct thread *td)
- int `ng_btsocket_rfcomm_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- int `ng_btsocket_rfcomm_control` (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- int `ng_btsocket_rfcomm_ctloutput` (struct socket *so, struct sockopt *sopt)
- void `ng_btsocket_rfcomm_detach` (struct socket *so)
- int `ng_btsocket_rfcomm_disconnect` (struct socket *so)
- int `ng_btsocket_rfcomm_listen` (struct socket *so, int backlog, struct thread *td)
- int `ng_btsocket_rfcomm_peeraddr` (struct socket *so, struct sockaddr **nam)
- int `ng_btsocket_rfcomm_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *nam, struct mbuf *control, struct thread *td)
- int `ng_btsocket_rfcomm_sockaddr` (struct socket *so, struct sockaddr **nam)

Variables

- int `ifqmaxlen`
- static u_int32_t `ng_btsocket_rfcomm_debug_level`
- static u_int32_t `ng_btsocket_rfcomm_timo`
- task `ng_btsocket_rfcomm_task`

7.69.1 Define Documentation

7.69.1.1 #define ALOT 0x7fff

Definition at line 89 of file `ng_btsocket_rfcomm.c`.

Referenced by `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_uih()`, and `ng_btsocket_rfcomm_session_process_pcb()`.

7.69.1.2 #define M_NETGRAPH_BT SOCKET_RFCOMM M_NETGRAPH

Definition at line 69 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_attach(), ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_session_create(), and ng_btsocket_rfcomm_sessions_task().

7.69.1.3 #define NG_BT SOCKET_RFCOMM_ALERT

Value:

```
if (ng_btsocket_rfcomm_debug_level >= NG_BT SOCKET_ALERT_LEVEL) \  
    printf
```

Definition at line 85 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_session_accept(), and ng_btsocket_rfcomm_upcall().

7.69.1.4 #define NG_BT SOCKET_RFCOMM_ERR

Value:

```
if (ng_btsocket_rfcomm_debug_level >= NG_BT SOCKET_ERR_LEVEL) \  
    printf
```

Definition at line 81 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_uih(), ng_btsocket_rfcomm_send_credits(), ng_btsocket_rfcomm_session_accept(), ng_btsocket_rfcomm_session_connect(), ng_btsocket_rfcomm_session_receive(), and ng_btsocket_rfcomm_session_send().

7.69.1.5 #define NG_BT SOCKET_RFCOMM_INFO

Value:

```
if (ng_btsocket_rfcomm_debug_level >= NG_BT SOCKET_INFO_LEVEL) \  
    printf
```

Definition at line 73 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_dm(), ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_receive_uih(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_credits(), ng_btsocket_rfcomm_send_msc(), ng_btsocket_rfcomm_send_pn(), ng_btsocket_rfcomm_send_uih(), ng_btsocket_rfcomm_session_clean(), ng_btsocket_rfcomm_session_task(), and ng_btsocket_rfcomm_set_pn().

7.69.1.6 #define ng_btsocket_rfcomm_task_wakeup() taskqueue_enqueue(taskqueue_swi_giant, &ng_btsocket_rfcomm_task)

Definition at line 98 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_detach(), ng_btsocket_rfcomm_disconnect(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_process_timeout(), ng_btsocket_rfcomm_send(), ng_btsocket_rfcomm_session_connect(), and ng_btsocket_rfcomm_upcall().

7.69.1.7 #define NG_BT_SOCKET_RFCOMM_WARN

Value:

```
if (ng_btsocket_rfcomm_debug_level >= NG_BT_SOCKET_WARN_LEVEL) \
    printf
```

Definition at line 77 of file ng_btsocket_rfcomm.c.

Referenced by ng_btsocket_rfcomm_receive_disc(), ng_btsocket_rfcomm_receive_dm(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_sabm(), ng_btsocket_rfcomm_receive_ua(), ng_btsocket_rfcomm_receive_uih(), and ng_btsocket_rfcomm_session_accept().

7.69.2 Function Documentation

7.69.2.1 static LIST_HEAD (ng_btsocket_rfcomm_session) [static]

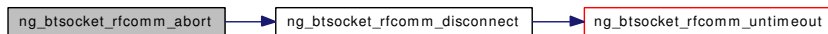
Definition at line 192 of file ng_btsocket_rfcomm.c.

7.69.2.2 void ng_btsocket_rfcomm_abort (struct socket * so)

Definition at line 347 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_disconnect().

Here is the call graph for this function:



7.69.2.3 int ng_btsocket_rfcomm_accept (struct socket * so, struct sockaddr ** nam)

Definition at line 367 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_peeraddr().

Here is the call graph for this function:



7.69.2.4 int ng_btsocket_rfcomm_attach (struct socket * so, int proto, struct thread * td)

Definition at line 377 of file ng_btsocket_rfcomm.c.

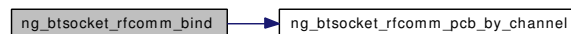
References `BLUETOOTH_PROTO_RFCOMM`, `M_NETGRAPH_BT SOCKET_RFCOMM`, `NG_BT SOCKET_RFCOMM_DLC_CFC`, `NG_BT SOCKET_RFCOMM_DLC_CLOSED`, `NG_BT SOCKET_RFCOMM_RECVSPACE`, `NG_BT SOCKET_RFCOMM_SENDSPEACE`, `RFCOMM_DEFAULT_CREDITS`, `RFCOMM_DEFAULT_MTU`, `RFCOMM_MODEM_DV`, `RFCOMM_MODEM_RTC`, `RFCOMM_MODEM_RTR`, and `so2rfcomm_pcb`.

7.69.2.5 int ng_btsocket_rfcomm_bind (struct socket * so, struct sockaddr * nam, struct thread * td)

Definition at line 440 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_pcb::channel`, `ng_btsocket_rfcomm_pcb_by_channel()`, `sockaddr_rfcomm::rfcomm_family`, `sockaddr_rfcomm::rfcomm_len`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::src`.

Here is the call graph for this function:

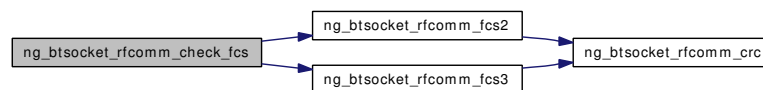
**7.69.2.6 static int ng_btsocket_rfcomm_check_fcs (u_int8_t * data, int type, u_int8_t fcs) [static]**

Definition at line 303 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_fcs2()`, `ng_btsocket_rfcomm_fcs3()`, and `RFCOMM_FRAME_UIH`.

Referenced by `ng_btsocket_rfcomm_receive_frame()`.

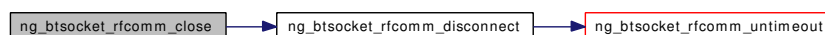
Here is the call graph for this function:

**7.69.2.7 void ng_btsocket_rfcomm_close (struct socket * so)**

Definition at line 355 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_disconnect()`.

Here is the call graph for this function:

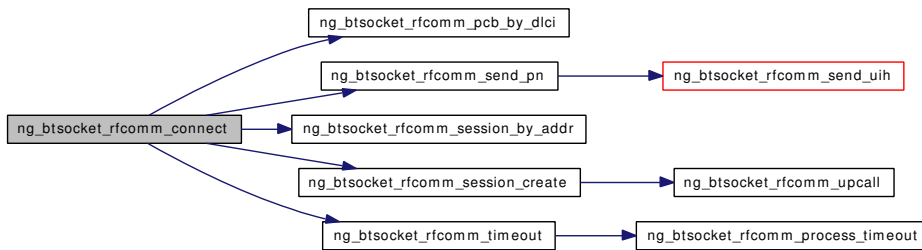


7.69.2.8 `int ng_btsocket_rfcomm_connect (struct socket * so, struct sockaddr * nam, struct thread * td)`

Definition at line 473 of file `ng_btsocket_rfcomm.c`.

References `BLUETOOTH_PROTO_L2CAP`, `ng_btsocket_rfcomm_pcb::channel`, `ng_btsocket_rfcomm_pcb::dlci`, `ng_btsocket_rfcomm_pcb::dst`, `INITIATOR`, `ng_btsocket_rfcomm_session::l2so`, `ng_btsocket_rfcomm_session::mtu`, `ng_btsocket_rfcomm_pcb::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING`, `NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT`, `ng_btsocket_rfcomm_pcb_by_dhci()`, `ng_btsocket_rfcomm_send_pn()`, `ng_btsocket_rfcomm_session_by_addr()`, `NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED`, `NG_BT_SOCKET_RFCOMM_SESSION_CONNECTING`, `ng_btsocket_rfcomm_session_create()`, `NG_BT_SOCKET_RFCOMM_SESSION_OPEN`, `ng_btsocket_rfcomm_task_wakeup`, `ng_btsocket_rfcomm_timeout()`, `NG_HCI_BDADDR_ANY`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `sockaddr_rfcomm::rfcomm_family`, `sockaddr_rfcomm::rfcomm_len`, `RFCOMM_MKDLCI`, `ng_btsocket_rfcomm_pcb::session`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::so`, `so2l2cap_pcb`, `so2rfcomm_pcb`, `ng_btsocket_rfcomm_pcb::src`, `ng_btsocket_rfcomm_pcb::state`, and `ng_btsocket_rfcomm_session::state`.

Here is the call graph for this function:



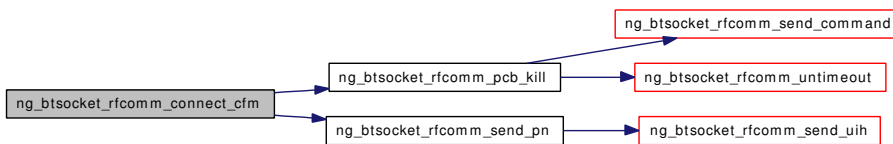
7.69.2.9 `static void ng_btsocket_rfcomm_connect_cfm (ng_btsocket_rfcomm_session_p s)` [static]

Definition at line 1163 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_session::l2so`, `ng_btsocket_rfcomm_session::mtu`, `ng_btsocket_rfcomm_pcb::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING`, `NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_send_pn()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_session::session_mtx`, `so2l2cap_pcb`, `ng_btsocket_rfcomm_pcb::src`, and `ng_btsocket_rfcomm_pcb::state`.

Referenced by `ng_btsocket_rfcomm_receive_sabm()`, and `ng_btsocket_rfcomm_receive_ua()`.

Here is the call graph for this function:



7.69.2.10 `static ng_btsocket_rfcomm_pcb_p ng_btsocket_rfcomm_connect_ind` (`ng_btsocket_rfcomm_session_p s`, `int channel`) [static]

Definition at line 1100 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_l2cap_pcb::dst`, `ng_btsocket_rfcomm_session::l2so`, `ng_btsocket_rfcomm_pcb_listener()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::so`, `so2l2cap_pcb`, `so2rfcomm_pcb`, and `ng_btsocket_l2cap_pcb::src`.

Referenced by `ng_btsocket_rfcomm_receive_pn()`, and `ng_btsocket_rfcomm_receive_sabm()`.

Here is the call graph for this function:



7.69.2.11 `int ng_btsocket_rfcomm_control` (`struct socket * so`, `u_long cmd`, `caddr_t data`, `struct ifnet * ifp`, `struct thread * td`)

Definition at line 613 of file `ng_btsocket_rfcomm.c`.

7.69.2.12 `static u_int8_t ng_btsocket_rfcomm_crc` (`u_int8_t * data`, `int length`) [static]

Definition at line 260 of file `ng_btsocket_rfcomm.c`.

Referenced by `ng_btsocket_rfcomm_fcs2()`, and `ng_btsocket_rfcomm_fcs3()`.

7.69.2.13 `int ng_btsocket_rfcomm_ctloutput` (`struct socket * so`, `struct sockopt * sopt`)

Definition at line 624 of file `ng_btsocket_rfcomm.c`.

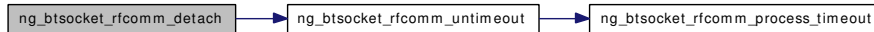
References `ng_btsocket_rfcomm_fc_info::cfc`, `ng_btsocket_rfcomm_pcb::flags`, `ng_btsocket_rfcomm_pcb::lmodem`, `ng_btsocket_rfcomm_fc_info::lmodem`, `ng_btsocket_rfcomm_pcb::mtu`, `NG_BTSOCKET_RFCOMM_DLC_CFC`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_fc_info::reserved`, `ng_btsocket_rfcomm_pcb::rmodem`, `ng_btsocket_rfcomm_fc_info::rmodem`, `ng_btsocket_rfcomm_pcb::rx_cred`, `ng_btsocket_rfcomm_fc_info::rx_cred`, `so2rfcomm_pcb`, `SO_RFCOMM_FC_INFO`, `SO_RFCOMM_MTU`, `SOL_RFCOMM`, `ng_btsocket_rfcomm_pcb::tx_cred`, and `ng_btsocket_rfcomm_fc_info::tx_cred`.

7.69.2.14 `void ng_btsocket_rfcomm_detach` (`struct socket * so`)

Definition at line 685 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::flags`, `M_NETGRAPH_BTSOCKET_RFCOMM`, `NG_BTSOCKET_RFCOMM_DLC_CLOSED`, `NG_BTSOCKET_RFCOMM_DLC_CONFIGURING`, `NG_BTSOCKET_RFCOMM_DLC_CONNECTED`, `NG_BTSOCKET_RFCOMM_DLC_CONNECTING`, `NG_BTSOCKET_RFCOMM_DLC_DETACHED`, `NG_BTSOCKET_RFCOMM_DLC_DISCONNECTING`, `NG_BTSOCKET_RFCOMM_DLC_TIMO`, `NG_BTSOCKET_RFCOMM_DLC_W4_CONNECT`, `ng_btsocket_rfcomm_task_wakeup`, `ng_btsocket_rfcomm_untimeout()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_pcb::session`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::state`.

Here is the call graph for this function:



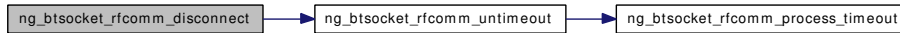
7.69.2.15 int ng_btsocket_rfcomm_disconnect (struct socket * so)

Definition at line 743 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::flags, NG_BTsocket_RFCOMM_DLC_CLOSED, NG_BTsocket_RFCOMM_DLC_CONFIGURING, NG_BTsocket_RFCOMM_DLC_CONNECTED, NG_BTsocket_RFCOMM_DLC_CONNECTING, NG_BTsocket_RFCOMM_DLC_DISCONNECTING, NG_BTsocket_RFCOMM_DLC_TIMO, NG_BTsocket_RFCOMM_DLC_W4_CONNECT, ng_btsocket_rfcomm_task_wakeup, ng_btsocket_rfcomm_untimeout(), ng_btsocket_rfcomm_pcb::pcb_mtx, so2rfcomm_pcb, and ng_btsocket_rfcomm_pcb::state.

Referenced by ng_btsocket_rfcomm_abort(), and ng_btsocket_rfcomm_close().

Here is the call graph for this function:



7.69.2.16 static u_int8_t ng_btsocket_rfcomm_fcs2 (u_int8_t * data) [static]

Definition at line 272 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_crc().

Referenced by ng_btsocket_rfcomm_check_fcs(), and ng_btsocket_rfcomm_send_uih().

Here is the call graph for this function:



7.69.2.17 static u_int8_t ng_btsocket_rfcomm_fcs3 (u_int8_t * data) [static]

Definition at line 279 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_crc().

Referenced by ng_btsocket_rfcomm_check_fcs(), and ng_btsocket_rfcomm_send_command().

Here is the call graph for this function:

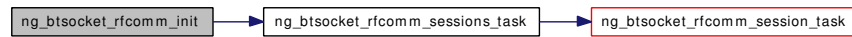


7.69.2.18 void ng_btsocket_rfcomm_init (void)

Definition at line 322 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_debug_level`, `ng_btsocket_rfcomm_sessions_task()`, `ng_btsocket_rfcomm_task`, `ng_btsocket_rfcomm_timo`, and `NG_BT_SOCKET_WARN_LEVEL`.

Here is the call graph for this function:

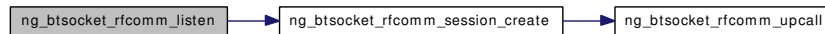


7.69.2.19 `int ng_btsocket_rfcomm_listen (struct socket * so, int backlog, struct thread * td)`

Definition at line 797 of file `ng_btsocket_rfcomm.c`.

References `BLUETOOTH_PROTO_L2CAP`, `ng_btsocket_rfcomm_pcb::channel`, `ng_btsocket_rfcomm_session_create()`, `NG_BT_SOCKET_RFCOMM_SESSION_LISTENING`, `NG_HCI_BDADDR_ANY`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_session::state`.

Here is the call graph for this function:



7.69.2.20 `static ng_btsocket_rfcomm_pcb_p ng_btsocket_rfcomm_pcb_by_channel (bdaddr_p src, int channel)` [static]

Definition at line 3348 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::channel`, and `ng_btsocket_rfcomm_pcb::src`.

Referenced by `ng_btsocket_rfcomm_bind()`.

7.69.2.21 `static ng_btsocket_rfcomm_pcb_p ng_btsocket_rfcomm_pcb_by_dhci (ng_btsocket_rfcomm_session_p s, int dhci)` [static]

Definition at line 3369 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::dhci`, and `ng_btsocket_rfcomm_session::session_mtx`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, and `ng_btsocket_rfcomm_receive_uih()`.

7.69.2.22 `static void ng_btsocket_rfcomm_pcb_kill (ng_btsocket_rfcomm_pcb_p pcb, int error)` [static]

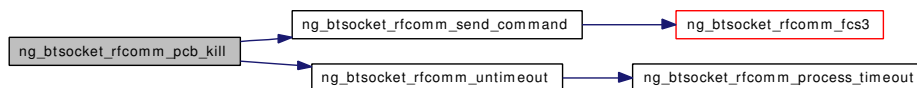
Definition at line 3273 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::dhci`, `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_rfcomm_pcb::flags`, `INITIATOR`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CLOSED`, `NG_BT_SOCKET_RFCOMM_DLC_TIMO`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_send_command()`, `NG_BT_SOCKET_RFCOMM_SESSION_CLOSED`, `NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED`, and `NG_BT_SOCKET_RFCOMM_SESSION_`

CONNECTING, NG_BT_SOCKET_RFCOMM_SESSION_DISCONNECTING, NG_BT_SOCKET_RFCOMM_SESSION_OPEN, ng_btsocket_rfcmm_task_wakeup, ng_btsocket_rfcmm_untimeout(), ng_btsocket_rfcmm_pcb::pcb_mtx, RFCOMM_FRAME_DISC, ng_btsocket_rfcmm_pcb::session, ng_btsocket_rfcmm_session::session_mtx, ng_btsocket_rfcmm_pcb::so, ng_btsocket_rfcmm_session::state, and ng_btsocket_rfcmm_pcb::state.

Referenced by ng_btsocket_rfcmm_connect_cfm(), ng_btsocket_rfcmm_receive_disc(), ng_btsocket_rfcmm_receive_dm(), ng_btsocket_rfcmm_receive_pn(), ng_btsocket_rfcmm_receive_sabm(), ng_btsocket_rfcmm_receive_ua(), ng_btsocket_rfcmm_session_clean(), and ng_btsocket_rfcmm_session_process_pcb().

Here is the call graph for this function:



7.69.2.23 static ng_btsocket_rfcmm_pcb_p ng_btsocket_rfcmm_pcb_listener (bdaddr_p src, int channel) [static]

Definition at line 3387 of file ng_btsocket_rfcmm.c.

References ng_btsocket_rfcmm_pcb::channel, NG_HCI_BDADDR_ANY, ng_btsocket_rfcmm_pcb::so, and ng_btsocket_rfcmm_pcb::src.

Referenced by ng_btsocket_rfcmm_connect_ind().

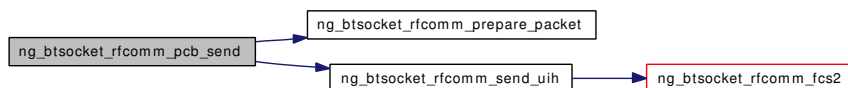
7.69.2.24 static int ng_btsocket_rfcmm_pcb_send (ng_btsocket_rfcmm_pcb_p pcb, int limit) [static]

Definition at line 3209 of file ng_btsocket_rfcmm.c.

References ng_btsocket_rfcmm_pcb::dlci, ng_btsocket_rfcmm_pcb::flags, INITIATOR, min, ng_btsocket_rfcmm_pcb::mtu, NG_BT_SOCKET_RFCOMM_DLC_CFC, NG_BT_SOCKET_RFCOMM_DLC_SENDING, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcmm_prepare_packet(), ng_btsocket_rfcmm_send_uih(), ng_btsocket_rfcmm_pcb::pcb_mtx, RFCOMM_MAX_CREDITS, RFCOMM_MKADDRESS, RFCOMM_MODEM_FC, ng_btsocket_rfcmm_pcb::rmodem, ng_btsocket_rfcmm_pcb::session, ng_btsocket_rfcmm_session::session_mtx, ng_btsocket_rfcmm_pcb::so, and ng_btsocket_rfcmm_pcb::tx_cred.

Referenced by ng_btsocket_rfcmm_receive_msc(), ng_btsocket_rfcmm_receive_uih(), and ng_btsocket_rfcmm_session_process_pcb().

Here is the call graph for this function:



7.69.2.25 int ng_btsocket_rfcmm_peeraddr (struct socket * so, struct sockaddr ** nam)

Definition at line 881 of file ng_btsocket_rfcmm.c.

References `ng_btsocket_rfcomm_pcb::channel`, `ng_btsocket_rfcomm_pcb::dst`, `sockaddr_rfcomm::rfcomm_bdaddr`, `sockaddr_rfcomm::rfcomm_channel`, `sockaddr_rfcomm::rfcomm_family`, `sockaddr_rfcomm::rfcomm_len`, and `so2rfcomm_pcb`.

Referenced by `ng_btsocket_rfcomm_accept()`.

7.69.2.26 `static struct mbuf * ng_btsocket_rfcomm_prepare_packet (struct sockbuf * sb, int length)` [static]

Definition at line 3496 of file `ng_btsocket_rfcomm.c`.

References `min`, and `NG_FREE_M`.

Referenced by `ng_btsocket_rfcomm_pcb_send()`.

7.69.2.27 `static void ng_btsocket_rfcomm_process_timeout (void * xpcb)` [static]

Definition at line 3456 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::dcli`, `ng_btsocket_rfcomm_pcb::flags`, `NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTING`, `NG_BT_SOCKET_RFCOMM_DLC_DISCONNECTING`, `NG_BT_SOCKET_RFCOMM_DLC_TIMEDOUT`, `NG_BT_SOCKET_RFCOMM_DLC_TIMO`, `NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_task_wakeup`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_pcb::so`, and `ng_btsocket_rfcomm_pcb::state`.

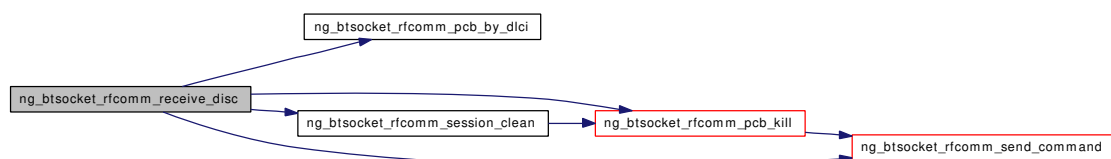
Referenced by `ng_btsocket_rfcomm_timeout()`, and `ng_btsocket_rfcomm_untimeout()`.

7.69.2.28 `static int ng_btsocket_rfcomm_receive_disc (ng_btsocket_rfcomm_session_p s, int dcli)` [static]

Definition at line 2049 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::flags`, `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_pcb_by_dcli()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_session_clean()`, `NG_BT_SOCKET_RFCOMM_SESSION_CLOSED`, `NG_BT_SOCKET_RFCOMM_SESSION_DISCONNECTING`, `NG_BT_SOCKET_RFCOMM_WARN`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `RFCOMM_FRAME_DM`, `RFCOMM_FRAME_UA`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::state`, and `ng_btsocket_rfcomm_session::state`.

Here is the call graph for this function:

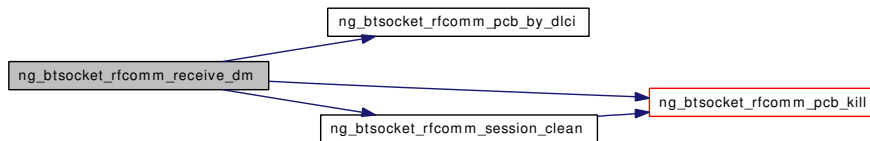


7.69.2.29 `static int ng_btsocket_rfcomm_receive_dm (ng_btsocket_rfcomm_session_p s, int dlc)`
`[static]`

Definition at line 2196 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::flags`, `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_pcb_by_dlc()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_session_clean()`, `NG_BT_SOCKET_RFCOMM_SESSION_CLOSED`, `NG_BT_SOCKET_RFCOMM_WARN`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::state`, and `ng_btsocket_rfcomm_session::state`.

Here is the call graph for this function:



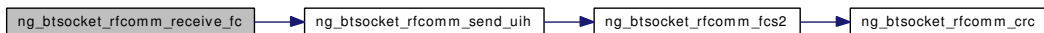
7.69.2.30 `static int ng_btsocket_rfcomm_receive_fc (ng_btsocket_rfcomm_session_p s, struct mbuf * m0)` `[static]`

Definition at line 2511 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_session::flags`, `INITIATOR`, `rfcomm_mcc_hdr::length`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_send_uih()`, `NG_BT_SOCKET_RFCOMM_SESSION_RFC`, `NG_FREE_M`, `RFCOMM_CR`, `RFCOMM_MCC_FCON`, `RFCOMM_MCC_LENGTH`, `RFCOMM_MCC_TYPE`, `RFCOMM_MKADDRESS`, `RFCOMM_MKMCC_TYPE`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_session::state`, and `rfcomm_mcc_hdr::type`.

Referenced by `ng_btsocket_rfcomm_receive_mcc()`.

Here is the call graph for this function:



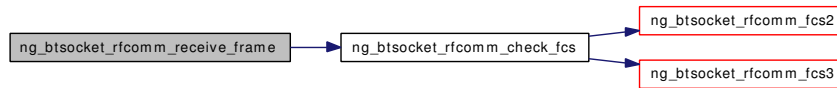
7.69.2.31 `static int ng_btsocket_rfcomm_receive_frame (ng_btsocket_rfcomm_session_p s, struct mbuf * m0)` `[static]`

Definition at line 1795 of file `ng_btsocket_rfcomm.c`.

References `rfcomm_frame_hdr::address`, `rfcomm_frame_hdr::control`, `rfcomm_frame_hdr::length`, `min`, `NG_BT_SOCKET_RFCOMM_ALERT`, `ng_btsocket_rfcomm_check_fcs()`, `NG_BT_SOCKET_RFCOMM_ERR`, `NG_BT_SOCKET_RFCOMM_INFO`, `NG_FREE_M`, `RFCOMM_CR`, `RFCOMM_DLCI`, `RFCOMM_EA`, `RFCOMM_PF`, `RFCOMM_TYPE`, and `ng_btsocket_rfcomm_session::session_mtx`.

Referenced by `ng_btsocket_rfcomm_session_receive()`.

Here is the call graph for this function:

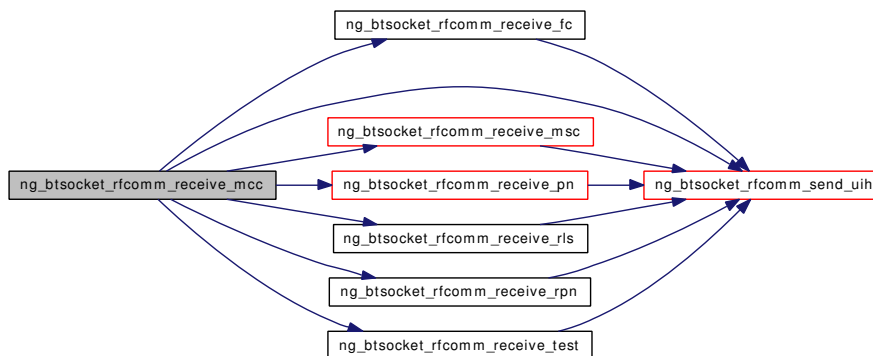


7.69.2.32 static int ng_btsocket_rfcomm_receive_mcc (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

Definition at line 2388 of file ng_btsocket_rfcomm.c.

References INITIATOR, rfcomm_mcc_hdr::length, NG_BT_SOCKET_RFCOMM_ERR, ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_send_uih(), NG_FREE_M, RFCOMM_CR, RFCOMM_MCC_FCOFF, RFCOMM_MCC_FCON, RFCOMM_MCC_LENGTH, RFCOMM_MCC_MSC, RFCOMM_MCC_NSC, RFCOMM_MCC_PN, RFCOMM_MCC_RLS, RFCOMM_MCC_RPN, RFCOMM_MCC_TEST, RFCOMM_MCC_TYPE, RFCOMM_MKADDRESS, RFCOMM_MKLEN8, RFCOMM_MKMCC_TYPE, ng_btsocket_rfcomm_session::session_mtx, and rfcomm_mcc_hdr::type.

Here is the call graph for this function:



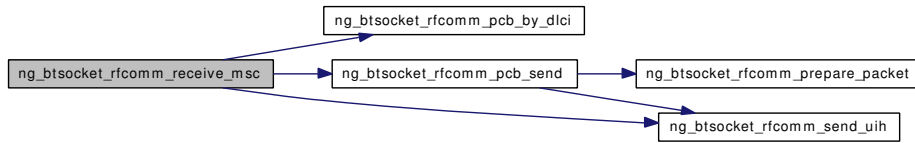
7.69.2.33 static int ng_btsocket_rfcomm_receive_msc (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

Definition at line 2551 of file ng_btsocket_rfcomm.c.

References rfcomm_mcc_msc::address, ALOT, ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_session::flags, INITIATOR, rfcomm_mcc_hdr::length, rfcomm_mcc_msc::modem, ng_btsocket_rfcomm_session::mtu, NG_BT_SOCKET_RFCOMM_DLC_CFC, NG_BT_SOCKET_RFCOMM_DLC_CONNECTED, NG_BT_SOCKET_RFCOMM_DLC_CONNECTING, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_pcb_by_dlc(), ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_send_uih(), NG_BT_SOCKET_RFCOMM_WARN, NG_FREE_M, ng_btsocket_rfcomm_pcb::pcb_mtx, RFCOMM_CR, RFCOMM_DLCI, RFCOMM_MCC_LENGTH, RFCOMM_MCC_MSC, RFCOMM_MKADDRESS, RFCOMM_MKMCC_TYPE, RFCOMM_MODEM_FC, ng_btsocket_rfcomm_pcb::rmodem, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_pcb::state, ng_btsocket_rfcomm_session::state, and rfcomm_mcc_hdr::type.

Referenced by ng_btsocket_rfcomm_receive_mcc().

Here is the call graph for this function:



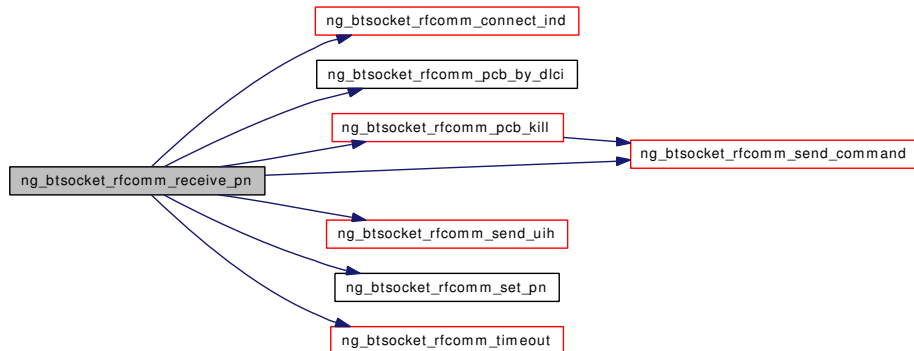
7.69.2.34 static int ng_btsocket_rfcomm_receive_pn (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

Definition at line 2767 of file ng_btsocket_rfcomm.c.

References rfcmm_mcc_pn::ack_timer, rfcmm_mcc_pn::credits, ng_btsocket_rfcomm_pcb::dlci, rfcmm_mcc_pn::dlci, ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_session::flags, rfcmm_mcc_pn::flow_control, INITIATOR, rfcmm_mcc_hdr::length, rfcmm_mcc_pn::max_retrans, ng_btsocket_rfcomm_session::mtu, rfcmm_mcc_pn::mtu, ng_btsocket_rfcomm_connect_ind(), NG_BT_SOCKET_RFCOMM_DLC_CFC, NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING, NG_BT_SOCKET_RFCOMM_DLC_CONNECTING, NG_BT_SOCKET_RFCOMM_ERR, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_pcb_by_dhci(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_uih(), ng_btsocket_rfcomm_set_pn(), ng_btsocket_rfcomm_timeout(), NG_BT_SOCKET_RFCOMM_WARN, NG_FREE_M, ng_btsocket_rfcomm_pcb::pcb_mtx, rfcmm_mcc_pn::priority, RFCOMM_CR, RFCOMM_DEFAULT_CREDITS, RFCOMM_FRAME_DM, RFCOMM_FRAME_SABM, RFCOMM_MCC_LENGTH, RFCOMM_MCC_PN, RFCOMM_MKADDRESS, RFCOMM_MKMCC_TYPE, RFCOMM_SRVCHANNEL, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_pcb::so, ng_btsocket_rfcomm_pcb::state, ng_btsocket_rfcomm_session::state, and rfcmm_mcc_hdr::type.

Referenced by ng_btsocket_rfcomm_receive_mcc().

Here is the call graph for this function:



7.69.2.35 static int ng_btsocket_rfcomm_receive_ri (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

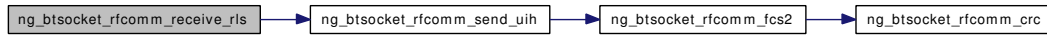
Definition at line 2726 of file ng_btsocket_rfcomm.c.

References rfcmm_mcc_ri::address, ng_btsocket_rfcomm_session::flags, INITIATOR, rfcmm_mcc_hdr::length, ng_btsocket_rfcomm_session::mtu, NG_BT_SOCKET_RFCOMM_ERR, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_send_uih(), NG_FREE_M, RFCOMM_CR, RFCOMM_DLCI, RFCOMM_MCC_LENGTH, RFCOMM_MCC_RLS, RFCOMM_MKADDRESS, RFCOMM_

MKMCC_TYPE, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_session::state, rfcomm_mcc_rls::status, and rfcomm_mcc_hdr::type.

Referenced by ng_btsocket_rfcomm_receive_mcc().

Here is the call graph for this function:



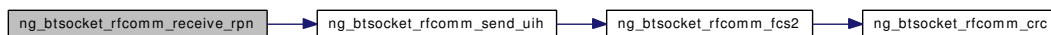
7.69.2.36 static int ng_btsocket_rfcomm_receive_rpn (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

Definition at line 2619 of file ng_btsocket_rfcomm.c.

References rfcomm_mcc_rpn::bit_rate, rfcomm_mcc_rpn::dlci, ng_btsocket_rfcomm_session::flags, rfcomm_mcc_rpn::flow_control, INITIATOR, rfcomm_mcc_hdr::length, rfcomm_mcc_rpn::line_settings, ng_btsocket_rfcomm_session::mtu, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_send_uih(), NG_FREE_M, rfcomm_mcc_rpn::param_mask, RFCOMM_CR, RFCOMM_DLCL, RFCOMM_MCC_LENGTH, RFCOMM_MCC_RPN, RFCOMM_MKADDRESS, RFCOMM_MKMCC_TYPE, RFCOMM_MKRPN_LINE_SETTINGS, RFCOMM_RPN_BR_115200, RFCOMM_RPN_DATA_8, RFCOMM_RPN_DATA_BITS, RFCOMM_RPN_FLOW_NONE, RFCOMM_RPN_PARITY, RFCOMM_RPN_PARITY_NONE, RFCOMM_RPN_PM_ALL, RFCOMM_RPN_PM_DATA, RFCOMM_RPN_PM_FLOW, RFCOMM_RPN_PM_PARITY, RFCOMM_RPN_PM_STOP, RFCOMM_RPN_PM_XOFF, RFCOMM_RPN_PM_XON, RFCOMM_RPN_STOP_1, RFCOMM_RPN_STOP_BITS, RFCOMM_RPN_XOFF_CHAR, RFCOMM_RPN_XON_CHAR, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_session::state, rfcomm_mcc_hdr::type, rfcomm_mcc_rpn::xoff_char, and rfcomm_mcc_rpn::xon_char.

Referenced by ng_btsocket_rfcomm_receive_mcc().

Here is the call graph for this function:

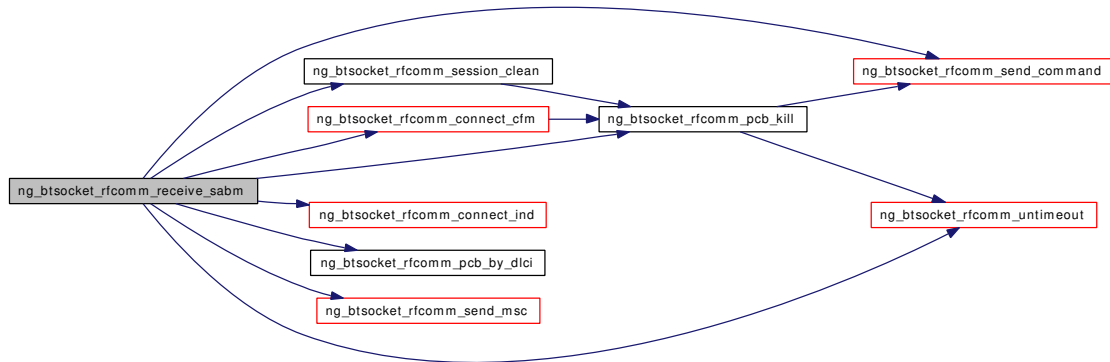


7.69.2.37 static int ng_btsocket_rfcomm_receive_sabm (ng_btsocket_rfcomm_session_p s, int dlci) [static]

Definition at line 1933 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::dlci, ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_session::flags, ng_btsocket_rfcomm_session::mtu, ng_btsocket_rfcomm_connect_cfm(), ng_btsocket_rfcomm_connect_ind(), NG_BT_SOCKET_RFCOMM_DLC_CONNECTED, NG_BT_SOCKET_RFCOMM_DLC_CONNECTING, NG_BT_SOCKET_RFCOMM_ERR, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_pcb_by_dlc(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_msc(), ng_btsocket_rfcomm_session_clean(), NG_BT_SOCKET_RFCOMM_SESSION_CLOSED, NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED, NG_BT_SOCKET_RFCOMM_SESSION_OPEN, ng_btsocket_rfcomm_untimeout(), NG_BT_SOCKET_RFCOMM_WARN, ng_btsocket_rfcomm_pcb::pcb_mtx, RFCOMM_FRAME_DM, RFCOMM_FRAME_UA, RFCOMM_SRVCHANNEL, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_pcb::so, ng_btsocket_rfcomm_pcb::state, and ng_btsocket_rfcomm_session::state.

Here is the call graph for this function:



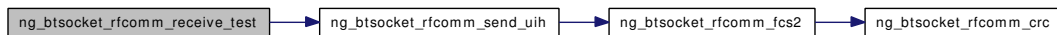
7.69.2.38 static int ng_btsocket_rfcomm_receive_test (ng_btsocket_rfcomm_session_p s, struct mbuf * m0) [static]

Definition at line 2484 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_session::flags, INITIATOR, rfcomm_mcc_hdr::length, ng_btsocket_rfcomm_session::mtu, NG_BTSOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_send_uih(), NG_FREE_M, RFCOMM_CR, RFCOMM_MCC_LENGTH, RFCOMM_MCC_TEST, RFCOMM_MKADDRESS, RFCOMM_MKMCC_TYPE, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_session::state, and rfcomm_mcc_hdr::type.

Referenced by ng_btsocket_rfcomm_receive_mcc().

Here is the call graph for this function:

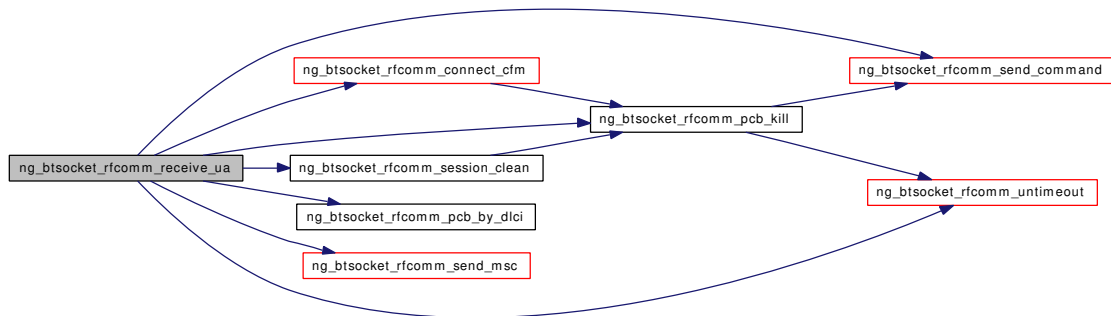


7.69.2.39 static int ng_btsocket_rfcomm_receive_ua (ng_btsocket_rfcomm_session_p s, int dlc) [static]

Definition at line 2112 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_session::flags, INITIATOR, ng_btsocket_rfcomm_session::mtu, ng_btsocket_rfcomm_connect_cfm(), NG_BTSOCKET_RFCOMM_DLC_CONNECTED, NG_BTSOCKET_RFCOMM_DLC_CONNECTING, NG_BTSOCKET_RFCOMM_DLC_DISCONNECTING, NG_BTSOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_pcb_by_dlc(), ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_send_msc(), ng_btsocket_rfcomm_session_clean(), NG_BTSOCKET_RFCOMM_SESSION_CLOSED, NG_BTSOCKET_RFCOMM_SESSION_CONNECTED, NG_BTSOCKET_RFCOMM_SESSION_DISCONNECTING, NG_BTSOCKET_RFCOMM_SESSION_OPEN, ng_btsocket_rfcomm_untimeout(), NG_BTSOCKET_RFCOMM_WARN, ng_btsocket_rfcomm_pcb::pcb_mtx, RFCOMM_FRAME_DM, ng_btsocket_rfcomm_session::session_mtx, ng_btsocket_rfcomm_pcb::so, ng_btsocket_rfcomm_pcb::state, and ng_btsocket_rfcomm_session::state.

Here is the call graph for this function:

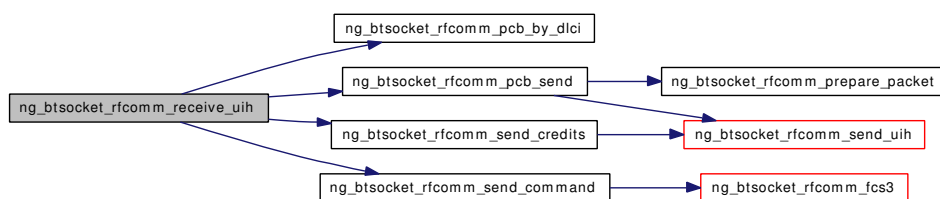


7.69.2.40 `static int ng_btsocket_rfcomm_receive_uh (ng_btsocket_rfcomm_session_p s, int dhci, int pf, struct mbuf * m0)` [static]

Definition at line 2242 of file ng_btsocket_rfcomm.c.

References `ALOT`, `ng_btsocket_rfcomm_pcb::flags`, `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_rfcomm_pcb::lmodem`, `ng_btsocket_rfcomm_pcb::mtu`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CFC`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED`, `NG_BT_SOCKET_RFCOMM_ERR`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_pcb_by_dhci()`, `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_send_command()`, `ng_btsocket_rfcomm_send_credits()`, `NG_BT_SOCKET_RFCOMM_SESSION_LFC`, `NG_BT_SOCKET_RFCOMM_WARN`, `NG_FREE_M`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `RFCOMM_FRAME_DM`, `RFCOMM_MAX_CREDITS`, `RFCOMM_MODEM_FC`, `ng_btsocket_rfcomm_pcb::rx_cred`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::so`, `ng_btsocket_rfcomm_pcb::state`, `ng_btsocket_rfcomm_session::state`, and `ng_btsocket_rfcomm_pcb::tx_cred`.

Here is the call graph for this function:



7.69.2.41 `int ng_btsocket_rfcomm_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * nam, struct mbuf * control, struct thread * td)`

Definition at line 904 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_pcb::flags`, `NG_BT_SOCKET_RFCOMM_DLC_CONNECTED`, `NG_BT_SOCKET_RFCOMM_DLC_SENDING`, `ng_btsocket_rfcomm_task_wakeup`, `NG_FREE_M`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_pcb::so`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::state`.

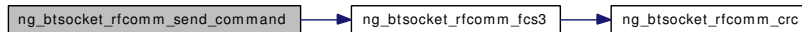
7.69.2.42 `static int ng_btsocket_rfcomm_send_command (ng_btsocket_rfcomm_session_p s, u_int8_t type, u_int8_t dlc)` [static]

Definition at line 2936 of file `ng_btsocket_rfcomm.c`.

References `rfcomm_cmd_hdr::address`, `rfcomm_cmd_hdr::control`, `rfcomm_cmd_hdr::fcs`, `ng_btsocket_rfcomm_session::flags`, `INITIATOR`, `rfcomm_cmd_hdr::length`, `ng_btsocket_rfcomm_session::mtu`, `NG_BT_MBUFQ_ENQUEUE`, `ng_btsocket_rfcomm_fcs3()`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_session::outq`, `RFCOMM_FRAME_DISC`, `RFCOMM_FRAME_DM`, `RFCOMM_FRAME_SABM`, `RFCOMM_FRAME_UA`, `RFCOMM_MKADDRESS`, `RFCOMM_MKCONTROL`, `RFCOMM_MKLEN8`, `ng_btsocket_rfcomm_session::session_mtx`, and `ng_btsocket_rfcomm_session::state`.

Referenced by `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, `ng_btsocket_rfcomm_receive_uih()`, `ng_btsocket_rfcomm_session_connect()`, and `ng_btsocket_rfcomm_session_process_pcb()`.

Here is the call graph for this function:



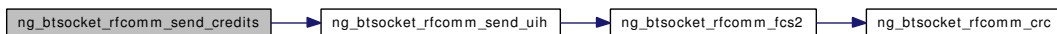
7.69.2.43 `static int ng_btsocket_rfcomm_send_credits (ng_btsocket_rfcomm_pcb_p pcb)` [static]

Definition at line 3155 of file `ng_btsocket_rfcomm.c`.

References `rfcomm_mcc_pn::credits`, `ng_btsocket_rfcomm_pcb::dlci`, `ng_btsocket_rfcomm_pcb::flags`, `INITIATOR`, `ng_btsocket_rfcomm_pcb::mtu`, `NG_BT_SOCKET_RFCOMM_ERR`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_send_uih()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `RFCOMM_MAX_CREDITS`, `RFCOMM_MKADDRESS`, `ng_btsocket_rfcomm_pcb::rx_cred`, `ng_btsocket_rfcomm_pcb::session`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::so`, `ng_btsocket_rfcomm_pcb::state`, and `ng_btsocket_rfcomm_pcb::tx_cred`.

Referenced by `ng_btsocket_rfcomm_receive_uih()`.

Here is the call graph for this function:



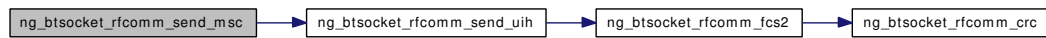
7.69.2.44 `static int ng_btsocket_rfcomm_send_msc (ng_btsocket_rfcomm_pcb_p pcb)` [static]

Definition at line 3062 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::dlci`, `ng_btsocket_rfcomm_pcb::flags`, `INITIATOR`, `rfcomm_mcc_hdr::length`, `ng_btsocket_rfcomm_pcb::lmodem`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_send_uih()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `RFCOMM_MCC_MSC`, `RFCOMM_MKADDRESS`, `RFCOMM_MKLEN8`, `RFCOMM_MKMCC_TYPE`, `ng_btsocket_rfcomm_pcb::session`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::state`, and `rfcomm_mcc_hdr::type`.

Referenced by `ng_btsocket_rfcomm_receive_sabm()`, and `ng_btsocket_rfcomm_receive_ua()`.

Here is the call graph for this function:



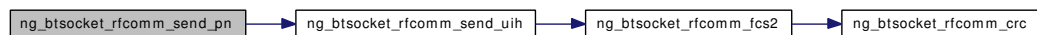
7.69.2.45 static int ng_btsocket_rfcomm_send_pn (ng_btsocket_rfcomm_pcb_p pcb) [static]

Definition at line 3100 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_pcb::dlci`, `ng_btsocket_rfcomm_pcb::flags`, `INITIATOR`, `rfcomm_mcc_hdr::length`, `ng_btsocket_rfcomm_pcb::mtu`, `NG_BT_SOCKET_RFCOMM_DLC_CFC`, `NG_BT_SOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_send_uih()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `RFCOMM_MCC_PN`, `RFCOMM_MKADDRESS`, `RFCOMM_MKLEN8`, `RFCOMM_MKMCC_TYPE`, `ng_btsocket_rfcomm_pcb::rx_cred`, `ng_btsocket_rfcomm_pcb::session`, `ng_btsocket_rfcomm_session::session_mtx`, `ng_btsocket_rfcomm_pcb::state`, and `rfcomm_mcc_hdr::type`.

Referenced by `ng_btsocket_rfcomm_connect()`, and `ng_btsocket_rfcomm_connect_cfm()`.

Here is the call graph for this function:



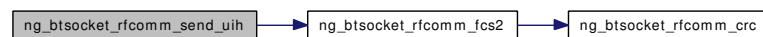
7.69.2.46 static int ng_btsocket_rfcomm_send_uih (ng_btsocket_rfcomm_session_p s, u_int8_t address, u_int8_t pf, u_int8_t credits, struct mbuf * data) [static]

Definition at line 2988 of file ng_btsocket_rfcomm.c.

References `rfcomm_frame_hdr::address`, `rfcomm_frame_hdr::control`, `ng_btsocket_rfcomm_session::flags`, `NG_BT_MBUFQ_ENQUEUE`, `ng_btsocket_rfcomm_fcs2()`, `NG_BT_SOCKET_RFCOMM_INFO`, `NG_FREE_M`, `ng_btsocket_rfcomm_session::outq`, `RFCOMM_FRAME_UIH`, `RFCOMM_MKCONTROL`, `RFCOMM_MKLEN16`, `RFCOMM_MKLEN8`, `ng_btsocket_rfcomm_session::session_mtx`, and `ng_btsocket_rfcomm_session::state`.

Referenced by `ng_btsocket_rfcomm_pcb_send()`, `ng_btsocket_rfcomm_receive_fc()`, `ng_btsocket_rfcomm_receive_mcc()`, `ng_btsocket_rfcomm_receive_msc()`, `ng_btsocket_rfcomm_receive_pn()`, `ng_btsocket_rfcomm_receive_rls()`, `ng_btsocket_rfcomm_receive_rpn()`, `ng_btsocket_rfcomm_receive_test()`, `ng_btsocket_rfcomm_send_credits()`, `ng_btsocket_rfcomm_send_msc()`, and `ng_btsocket_rfcomm_send_pn()`.

Here is the call graph for this function:



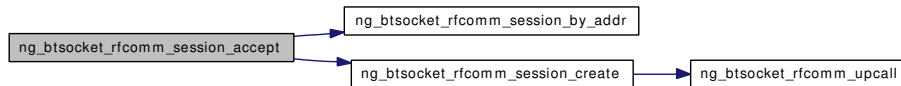
7.69.2.47 static int ng_btsocket_rfcomm_session_accept (ng_btsocket_rfcomm_session_p s0) [static]

Definition at line 1355 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_l2cap_pcb::dst`, `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_l2cap_pcb::imtu`, `ng_btsocket_rfcomm_session::l2so`, `min`, `ng_btsocket_rfcomm_session::mtu`, `NG_BTSOCKET_RFCOMM_ALERT`, `NG_BTSOCKET_RFCOMM_ERR`, `ng_btsocket_rfcomm_session_by_addr()`, `NG_BTSOCKET_RFCOMM_SESSION_CONNECTED`, `ng_btsocket_rfcomm_session_create()`, `NG_BTSOCKET_RFCOMM_WARN`, `ng_btsocket_l2cap_pcb::omtu`, `ng_btsocket_rfcomm_session::session_mtx`, `so2l2cap_pcb`, `ng_btsocket_l2cap_pcb::src`, and `ng_btsocket_rfcomm_session::state`.

Referenced by `ng_btsocket_rfcomm_session_task()`.

Here is the call graph for this function:



7.69.2.48 `static ng_btsocket_rfcomm_session_p ng_btsocket_rfcomm_session_by_addr (bdaddr_p src, bdaddr_p dst)` [static]

Definition at line 1762 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_l2cap_pcb::dst`, `ng_btsocket_rfcomm_session::l2so`, `NG_HCI_BDADDR_ANY`, `so2l2cap_pcb`, and `ng_btsocket_l2cap_pcb::src`.

Referenced by `ng_btsocket_rfcomm_connect()`, and `ng_btsocket_rfcomm_session_accept()`.

7.69.2.49 `static void ng_btsocket_rfcomm_session_clean (ng_btsocket_rfcomm_session_p s)` [static]

Definition at line 1628 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::dcli`, `ng_btsocket_rfcomm_pcb::flags`, `NG_BTSOCKET_RFCOMM_DLC_CONNECTED`, `NG_BTSOCKET_RFCOMM_INFO`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, `ng_btsocket_rfcomm_session::session_mtx`, and `ng_btsocket_rfcomm_pcb::state`.

Referenced by `ng_btsocket_rfcomm_receive_disc()`, `ng_btsocket_rfcomm_receive_dm()`, `ng_btsocket_rfcomm_receive_sabm()`, `ng_btsocket_rfcomm_receive_ua()`, and `ng_btsocket_rfcomm_session_task()`.

Here is the call graph for this function:



7.69.2.50 `static int ng_btsocket_rfcomm_session_connect (ng_btsocket_rfcomm_session_p s)` [static]

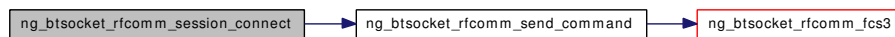
Definition at line 1465 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_session::flags`, `ng_btsocket_l2cap_pcb::imtu`, `ng_btsocket_rfcomm_session::l2so`, `min`, `ng_btsocket_rfcomm_session::mtu`, `NG_BTSOCKET_RFCOMM_ERR`, `ng_`

btsocket_rfcomm_send_command(), NG_BTsocket_RFCOMM_SESSION_CONNECTED, ng_btsocket_rfcomm_task_wakeup, ng_btsocket_l2cap_pcb::omtu, RFCOMM_FRAME_SABM, ng_btsocket_rfcomm_session::session_mtx, so2l2cap_pcb, and ng_btsocket_rfcomm_session::state.

Referenced by ng_btsocket_rfcomm_session_task().

Here is the call graph for this function:



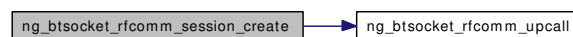
7.69.2.51 static int ng_btsocket_rfcomm_session_create (ng_btsocket_rfcomm_session_p * sp, struct socket * l2so, bdaddr_p src, bdaddr_p dst, struct thread * td) [static]

Definition at line 1211 of file ng_btsocket_rfcomm.c.

References ifqmaxlen, sockaddr_l2cap::l2cap_bdaddr, sockaddr_l2cap::l2cap_family, sockaddr_l2cap::l2cap_len, sockaddr_l2cap::l2cap_psm, M_NETGRAPH_BTsocket_RFCOMM, NG_BT_MBUFQ_INIT, NG_BTsocket_RFCOMM_SESSION_CLOSED, NG_BTsocket_RFCOMM_SESSION_CONNECTING, NG_BTsocket_RFCOMM_SESSION_INITIATOR, NG_BTsocket_RFCOMM_SESSION_LISTENING, ng_btsocket_rfcomm_upcall(), NG_L2CAP_PSM_RFCOMM, RFCOMM_DEFAULT_MTU, SO_L2CAP_IMTU, and SOL_L2CAP.

Referenced by ng_btsocket_rfcomm_connect(), ng_btsocket_rfcomm_listen(), and ng_btsocket_rfcomm_session_accept().

Here is the call graph for this function:



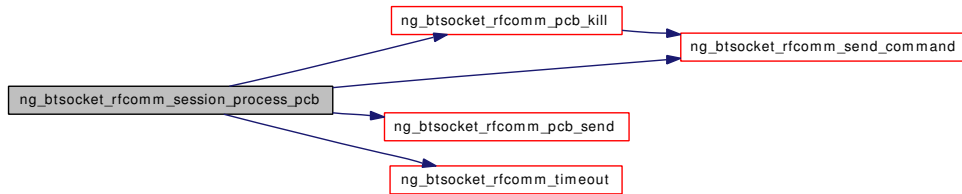
7.69.2.52 static void ng_btsocket_rfcomm_session_process_pcb (ng_btsocket_rfcomm_session_p s) [static]

Definition at line 1665 of file ng_btsocket_rfcomm.c.

References ALOT, ng_btsocket_rfcomm_pcb::flags, NG_BTsocket_RFCOMM_DLC_CONFIGURING, NG_BTsocket_RFCOMM_DLC_CONNECTED, NG_BTsocket_RFCOMM_DLC_CONNECTING, NG_BTsocket_RFCOMM_DLC_DETACHED, NG_BTsocket_RFCOMM_DLC_DISCONNECTING, NG_BTsocket_RFCOMM_DLC_TIMEDOUT, NG_BTsocket_RFCOMM_DLC_TIMO, NG_BTsocket_RFCOMM_DLC_W4_CONNECT, ng_btsocket_rfcomm_pcb_kill(), ng_btsocket_rfcomm_pcb_send(), ng_btsocket_rfcomm_send_command(), ng_btsocket_rfcomm_timeout(), ng_btsocket_rfcomm_pcb::pcb_mtx, RFCOMM_FRAME_DISC, ng_btsocket_rfcomm_session::session_mtx, and ng_btsocket_rfcomm_pcb::state.

Referenced by ng_btsocket_rfcomm_session_task().

Here is the call graph for this function:



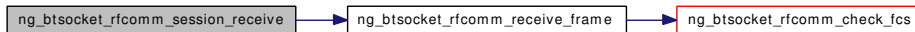
7.69.2.53 static int ng_btsocket_rfcomm_session_receive (ng_btsocket_rfcomm_session_p s) [static]

Definition at line 1518 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_session::flags, ng_btsocket_rfcomm_session::l2so, NG_BT_SOCKET_RFCOMM_ERR, ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_session::session_mtx, and ng_btsocket_rfcomm_session::state.

Referenced by ng_btsocket_rfcomm_session_task().

Here is the call graph for this function:



7.69.2.54 static int ng_btsocket_rfcomm_session_send (ng_btsocket_rfcomm_session_p s) [static]

Definition at line 1584 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_session::flags, ng_btsocket_rfcomm_session::l2so, NG_BT_MBUFQ_DEQUEUE, NG_BT_SOCKET_RFCOMM_ERR, ng_btsocket_rfcomm_session::outq, ng_btsocket_rfcomm_session::session_mtx, and ng_btsocket_rfcomm_session::state.

Referenced by ng_btsocket_rfcomm_session_task().

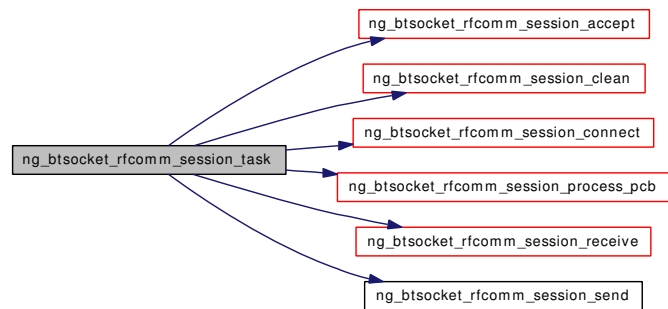
7.69.2.55 static void ng_btsocket_rfcomm_session_task (ng_btsocket_rfcomm_session_p s) [static]

Definition at line 1038 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_session::flags, ng_btsocket_rfcomm_session::l2so, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_session_accept(), ng_btsocket_rfcomm_session_clean(), NG_BT_SOCKET_RFCOMM_SESSION_CLOSED, ng_btsocket_rfcomm_session_connect(), NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED, NG_BT_SOCKET_RFCOMM_SESSION_CONNECTING, NG_BT_SOCKET_RFCOMM_SESSION_DISCONNECTING, NG_BT_SOCKET_RFCOMM_SESSION_LISTENING, NG_BT_SOCKET_RFCOMM_SESSION_OPEN, ng_btsocket_rfcomm_session_process_pcb(), ng_btsocket_rfcomm_session_receive(), ng_btsocket_rfcomm_session_send(), ng_btsocket_rfcomm_session::session_mtx, and ng_btsocket_rfcomm_session::state.

Referenced by ng_btsocket_rfcomm_sessions_task().

Here is the call graph for this function:



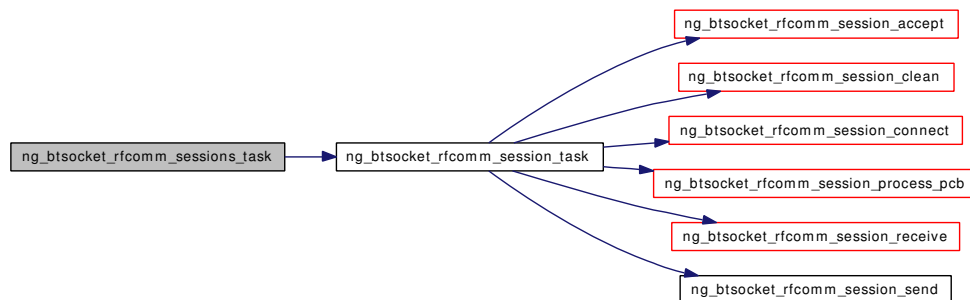
7.69.2.56 static void ng_btsocket_rfcomm_sessions_task (void * ctx, int pending) [static]

Definition at line 988 of file ng_btsocket_rfcomm.c.

References M_NETGRAPH_BT_SOCKET_RFCOMM, NG_BT_MBUFQ_DRAIN, NG_BT_SOCKET_RFCOMM_SESSION_CLOSED, ng_btsocket_rfcomm_session_task(), and ng_btsocket_rfcomm_session::session_mtx.

Referenced by ng_btsocket_rfcomm_init().

Here is the call graph for this function:



7.69.2.57 static void ng_btsocket_rfcomm_set_pn (ng_btsocket_rfcomm_pcb_p pcb, u_int8_t cr, u_int8_t flow_control, u_int8_t credits, u_int16_t mtu) [static]

Definition at line 2900 of file ng_btsocket_rfcomm.c.

References ng_btsocket_rfcomm_pcb::dlci, ng_btsocket_rfcomm_pcb::flags, ng_btsocket_rfcomm_pcb::mtu, NG_BT_SOCKET_RFCOMM_DLC_CFC, NG_BT_SOCKET_RFCOMM_INFO, ng_btsocket_rfcomm_pcb::pcb_mtx, ng_btsocket_rfcomm_pcb::rx_cred, ng_btsocket_rfcomm_pcb::state, and ng_btsocket_rfcomm_pcb::tx_cred.

Referenced by ng_btsocket_rfcomm_receive_pn().

7.69.2.58 int ng_btsocket_rfcomm_sockaddr (struct socket * so, struct sockaddr ** nam)

Definition at line 947 of file ng_btsocket_rfcomm.c.

References `ng_btsocket_rfcomm_pcb::channel`, `sockaddr_rfcomm::rfcomm_bdaddr`, `sockaddr_rfcomm::rfcomm_channel`, `sockaddr_rfcomm::rfcomm_family`, `sockaddr_rfcomm::rfcomm_len`, `so2rfcomm_pcb`, and `ng_btsocket_rfcomm_pcb::src`.

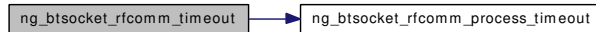
7.69.2.59 `static void ng_btsocket_rfcomm_timeout (ng_btsocket_rfcomm_pcb_p pcb)` [static]

Definition at line 3421 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::flags`, `NG_BT_SOCKET_RFCOMM_DLC_TIMEDOUT`, `NG_BT_SOCKET_RFCOMM_DLC_TIMO`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_timo`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, and `ng_btsocket_rfcomm_pcb::timo`.

Referenced by `ng_btsocket_rfcomm_connect()`, `ng_btsocket_rfcomm_receive_pn()`, and `ng_btsocket_rfcomm_session_process_pcb()`.

Here is the call graph for this function:



7.69.2.60 `static void ng_btsocket_rfcomm_untimeout (ng_btsocket_rfcomm_pcb_p pcb)` [static]

Definition at line 3439 of file `ng_btsocket_rfcomm.c`.

References `ng_btsocket_rfcomm_pcb::flags`, `NG_BT_SOCKET_RFCOMM_DLC_TIMEDOUT`, `NG_BT_SOCKET_RFCOMM_DLC_TIMO`, `ng_btsocket_rfcomm_process_timeout()`, `ng_btsocket_rfcomm_pcb::pcb_mtx`, and `ng_btsocket_rfcomm_pcb::timo`.

Referenced by `ng_btsocket_rfcomm_detach()`, `ng_btsocket_rfcomm_disconnect()`, `ng_btsocket_rfcomm_pcb_kill()`, `ng_btsocket_rfcomm_receive_sabm()`, and `ng_btsocket_rfcomm_receive_ua()`.

Here is the call graph for this function:



7.69.2.61 `static void ng_btsocket_rfcomm_upcall (struct socket * so, void * arg, int waitflag)` [static]

Definition at line 970 of file `ng_btsocket_rfcomm.c`.

References `NG_BT_SOCKET_RFCOMM_ALERT`, and `ng_btsocket_rfcomm_task_wakeup`.

Referenced by `ng_btsocket_rfcomm_session_create()`.

7.69.3 Variable Documentation

7.69.3.1 `int ifqmaxlen`

7.69.3.2 `u_int32_t ng_btsocket_rfcomm_debug_level` [static]

Definition at line 189 of file `ng_btsocket_rfcomm.c`.

Referenced by `ng_btsocket_rfcomm_init()`.

7.69.3.3 struct task `ng_btsocket_rfcomm_task`

Definition at line 191 of file `ng_btsocket_rfcomm.c`.

Referenced by `ng_btsocket_rfcomm_init()`.

7.69.3.4 `u_int32_t ng_btsocket_rfcomm_timo` [static]

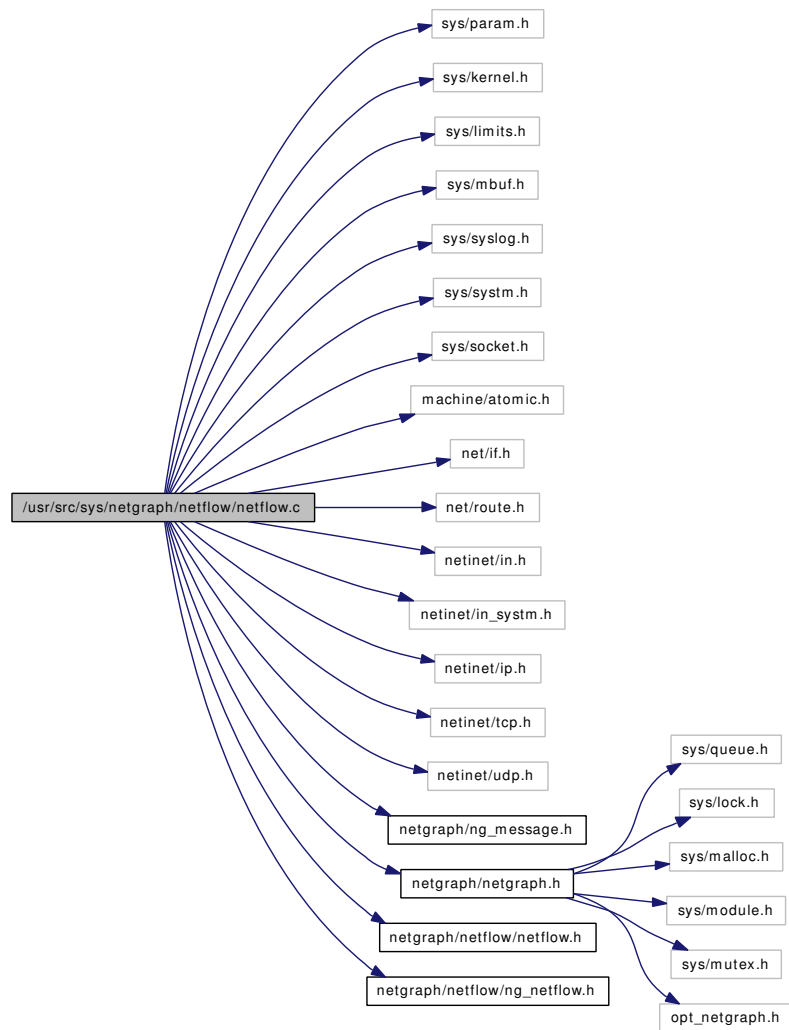
Definition at line 190 of file `ng_btsocket_rfcomm.c`.

Referenced by `ng_btsocket_rfcomm_init()`, and `ng_btsocket_rfcomm_timeout()`.

7.70 /usr/src/sys/netgraph/netflow/netflow.c File Reference

```
#include <sys/param.h>
#include <sys/kernel.h>
#include <sys/limits.h>
#include <sys/mbuf.h>
#include <sys/syslog.h>
#include <sys/system.h>
#include <sys/socket.h>
#include <machine/atomic.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <netinet/udp.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/netflow/netflow.h>
#include <netgraph/netflow/ng_netflow.h>
```

Include dependency graph for netflow.c:



Defines

- #define [NBUCKETS](#) (65536)
- #define [FULL_HASH](#)(addr1, addr2, port1, port2)
- #define [ADDR_HASH](#)(addr1, addr2)
- #define [INACTIVE](#)(fle) (time_uptime - fle → f.last > [priv](#) → info.nfinfo_inact_t)
- #define [AGED](#)(fle) (time_uptime - fle → f.first > [priv](#) → info.nfinfo_act_t)
- #define [ISFREE](#)(fle) (fle → f.packets == 0)
- #define [SMALL](#)(fle) (fle → f.packets <= 4)
- #define [MILLIuptime](#)(t)

Functions

- [MALLOC_DECLARE](#) (M_NETFLOW_HASH)
- [MALLOC_DEFINE](#) (M_NETFLOW_HASH, "netflow_hash", "NetFlow hash")
- static int [export_add](#) (item_p, struct [flow_entry](#) *)
- static int [export_send](#) (priv_p, item_p, int flags)

- static `__inline uint32_t ip_hash` (struct `flow_rec *r`)
- static int `uma_ctor_flow` (void *mem, int size, void *arg, int how)
- static void `uma_dtor_flow` (void *mem, int size, void *arg)
- static `item_p get_export_dgram` (priv_p priv)
- static void `return_export_dgram` (priv_p priv, item_p item, int flags)
- static `__inline void expire_flow` (priv_p priv, item_p *item, struct `flow_entry *fle`, int flags)
- void `ng_netflow_copyinfo` (priv_p priv, struct `ng_netflow_info *i`)
- static `__inline int hash_insert` (priv_p priv, struct `flow_hash_entry *hsh`, struct `flow_rec *r`, int plen, uint8_t tcp_flags)
- int `ng_netflow_cache_init` (priv_p priv)
- void `ng_netflow_cache_flush` (priv_p priv)
- int `ng_netflow_flow_add` (priv_p priv, struct ip *ip, iface_p iface, struct ifnet *ifp)
- int `ng_netflow_flow_show` (priv_p priv, uint32_t last, struct `ng_mesg *resp`)
- void `ng_netflow_expire` (void *arg)

Variables

- static const char `rcs_id` []

7.70.1 Define Documentation

7.70.1.1 #define ADDR_HASH(addr1, addr2)

Value:

```
((addr1 ^ (addr1 >> 16) ^
  htons(addr2 ^ (addr2 >> 16))) &
 (NBUCKETS - 1))
```

Definition at line 67 of file `netflow.c`.

Referenced by `ip_hash()`.

7.70.1.2 #define AGED(fle) (time_uptime - fle → f.first > priv → info.nfinfo_act_t)

Definition at line 75 of file `netflow.c`.

Referenced by `ng_netflow_expire()`, and `ng_netflow_flow_add()`.

7.70.1.3 #define FULL_HASH(addr1, addr2, port1, port2)

Value:

```
((((addr1 ^ (addr1 >> 16) ^
  htons(addr2 ^ (addr2 >> 16))) ^
  port1 ^ htons(port2)) &
 (NBUCKETS - 1))
```

Definition at line 60 of file `netflow.c`.

Referenced by `ip_hash()`.

7.70.1.4 #define INACTIVE(fle) (time_uptime - fle → f.last > priv → info.nfinfo_inact_t)

Definition at line 74 of file netflow.c.

Referenced by ng_netflow_expire(), and ng_netflow_flow_add().

7.70.1.5 #define ISFREE(fle) (fle → f.packets == 0)

Definition at line 76 of file netflow.c.

7.70.1.6 #define MILLIUPTIME(t)

Value:

```
((t) << 9) + /* 512 */ \
          ((t) << 8) + /* 256 */ \
          ((t) << 7) + /* 128 */ \
          ((t) << 6) + /* 64 */ \
          ((t) << 5) + /* 32 */ \
          ((t) << 3))
```

Definition at line 91 of file netflow.c.

Referenced by export_add(), and export_send().

7.70.1.7 #define NBUCKETS (65536)

Definition at line 57 of file netflow.c.

Referenced by ng_netflow_cache_flush(), ng_netflow_cache_init(), ng_netflow_expire(), and ng_netflow_flow_show().

7.70.1.8 #define SMALL(fle) (fle → f.packets <= 4)

Definition at line 84 of file netflow.c.

Referenced by ng_netflow_expire(), and ng_netflow_flow_add().

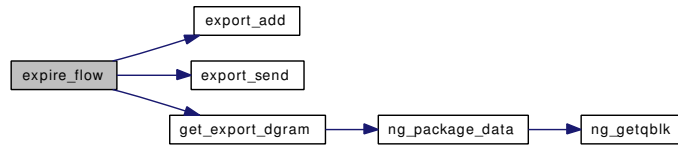
7.70.2 Function Documentation**7.70.2.1 static __inline void expire_flow (priv_p priv, item_p * item, struct flow_entry * fle, int flags) [static]**

Definition at line 202 of file netflow.c.

References export_add(), export_send(), and get_export_dgram().

Referenced by ng_netflow_cache_flush(), ng_netflow_expire(), and ng_netflow_flow_add().

Here is the call graph for this function:



7.70.2.2 static int export_add (item_p, struct flow_entry *) [static]

Definition at line 628 of file netflow.c.

References flow_entry_data::bytes, netflow_v5_header::count, netflow_v5_record::d_port, netflow_v5_record::dst_addr, netflow_v5_record::dst_as, flow_entry_data::dst_mask, netflow_v5_record::dst_mask, flow_entry::f, flow_entry_data::first, netflow_v5_record::first, netflow_v5_record::flags, flow_entry_data::fle_o_ifx, netflow_v5_export_dgram::header, netflow_v5_record::i_ifx, flow_entry_data::last, netflow_v5_record::last, MILLIuptime, NETFLOW_V5_MAX_RECORDS, flow_entry_data::next_hop, netflow_v5_record::next_hop, NGI_M, netflow_v5_record::o_ifx, netflow_v5_record::octets, flow_entry_data::packets, netflow_v5_record::packets, netflow_v5_record::prot, flow_entry_data::r, netflow_v5_export_dgram::r, flow_rec::r_dst, flow_rec::r_src, netflow_v5_record::s_port, netflow_v5_record::src_addr, netflow_v5_record::src_as, flow_entry_data::src_mask, netflow_v5_record::src_mask, flow_entry_data::tcp_flags, and netflow_v5_record::tos.

Referenced by expire_flow().

7.70.2.3 static int export_send (priv_p, item_p, int flags) [static]

Definition at line 593 of file netflow.c.

References netflow_v5_header::count, netflow_v5_header::engine_id, netflow_v5_header::engine_type, netflow_v5_header::flow_seq, netflow_v5_export_dgram::header, MILLIuptime, NG_FWD_ITEM_HOOK_FLAGS, NGI_M, netflow_v5_header::pad, netflow_v5_header::sys_uptime, netflow_v5_header::unix_nsecs, and netflow_v5_header::unix_secs.

Referenced by expire_flow(), ng_netflow_cache_flush(), and return_export_dgram().

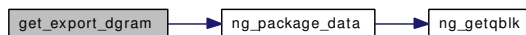
7.70.2.4 static item_p get_export_dgram (priv_p priv) [static]

Definition at line 146 of file netflow.c.

References netflow_v5_header::count, netflow_v5_export_dgram::header, NETFLOW_V5, NG_NOFLAGS, ng_package_data(), and netflow_v5_header::version.

Referenced by expire_flow().

Here is the call graph for this function:



7.70.2.5 static __inline int hash_insert (priv_p priv, struct flow_hash_entry * hsh, struct flow_rec * r, int plen, uint8_t tcp_flags) [static]

Definition at line 237 of file netflow.c.

References `flow_entry_data::bytes`, `flow_entry_data::dst_mask`, `flow_entry::f`, `flow_entry_data::first`, `flow_entry_data::fle_o_ifx`, `flow_entry_data::last`, `flow_hash_entry::mtx`, `flow_entry_data::next_hop`, `flow_entry_data::packets`, `flow_entry_data::r`, `flow_rec::r_dst`, `flow_rec::r_src`, `flow_entry_data::src_mask`, and `flow_entry_data::tcp_flags`.

Referenced by `ng_netflow_flow_add()`.

7.70.2.6 `static __inline uint32_t ip_hash (struct flow_rec * r) [static]`

Definition at line 106 of file `netflow.c`.

References `ADDR_HASH`, `FULL_HASH`, `flow_rec::r_dst`, and `flow_rec::r_src`.

Referenced by `ng_netflow_flow_add()`.

7.70.2.7 `MALLOC_DECLARE (M_NETFLOW_HASH)`

7.70.2.8 `MALLOC_DEFINE (M_NETFLOW_HASH, "netflow_hash", "NetFlow hash")`

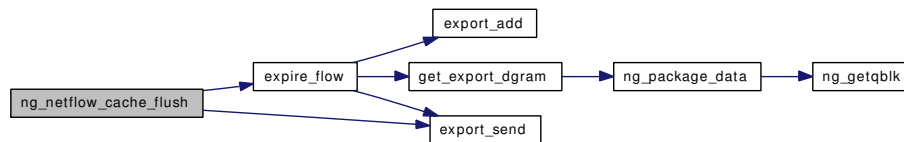
7.70.2.9 `void ng_netflow_cache_flush (priv_p priv)`

Definition at line 361 of file `netflow.c`.

References `expire_flow()`, `export_send()`, `flow_hash_entry::mtx`, `NBUCKETS`, and `NG_QUEUE`.

Referenced by `ng_netflow_close()`.

Here is the call graph for this function:



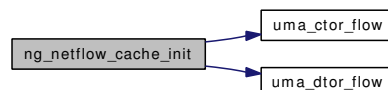
7.70.2.10 `int ng_netflow_cache_init (priv_p priv)`

Definition at line 328 of file `netflow.c`.

References `CACHESIZE`, `flow_hash_entry::mtx`, `NBUCKETS`, `uma_ctor_flow()`, and `uma_dtor_flow()`.

Referenced by `ng_netflow_constructor()`.

Here is the call graph for this function:



7.70.2.11 `void ng_netflow_copyinfo (priv_p priv, struct ng_netflow_info * i)`

Definition at line 220 of file `netflow.c`.

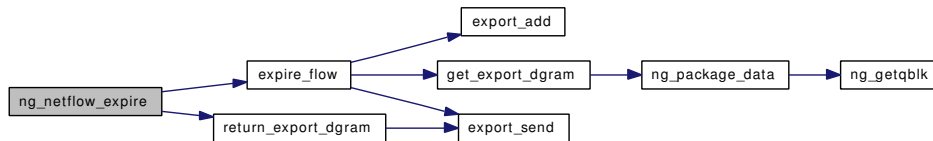
Referenced by `ng_netflow_rcvmsg()`.

7.70.2.12 void ng_netflow_expire (void * arg)

Definition at line 675 of file `netflow.c`.

References `AGED`, `expire_flow()`, `INACTIVE`, `flow_hash_entry::mtx`, `NBUCKETS`, `ng_netflow_expire`, `NG_NOFLAGS`, `return_export_dgram()`, and `SMALL`.

Here is the call graph for this function:



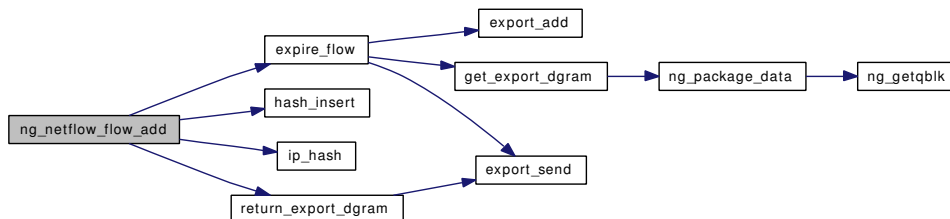
7.70.2.13 int ng_netflow_flow_add (priv_p priv, struct ip * ip, iface_p iface, struct ifnet * ifp)

Definition at line 397 of file `netflow.c`.

References `AGED`, `flow_entry_data::bytes`, `expire_flow()`, `flow_entry::f`, `hash_insert()`, `ng_netflow_ifinfo::ifinfo_index`, `INACTIVE`, `ng_netflow_iface::info`, `ip_hash()`, `flow_entry_data::last`, `flow_hash_entry::mtx`, `NG_QUEUE`, `flow_entry_data::packets`, `flow_entry_data::r`, `return_export_dgram()`, `SMALL`, and `flow_entry_data::tcp_flags`.

Referenced by `ng_netflow_rcvdata()`.

Here is the call graph for this function:



7.70.2.14 int ng_netflow_flow_show (priv_p priv, uint32_t last, struct ng_mesg * resp)

Definition at line 533 of file `netflow.c`.

References `ng_mesg::data`, `ngnf_flows::entries`, `ngnf_flows::last`, `flow_hash_entry::mtx`, `NBUCKETS`, `ngnf_flows::nentries`, and `NREC_AT_ONCE`.

Referenced by `ng_netflow_rcvmsg()`.

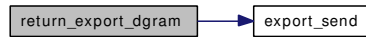
7.70.2.15 static void return_export_dgram (priv_p priv, item_p item, int flags) [static]

Definition at line 180 of file `netflow.c`.

References `export_send()`.

Referenced by `ng_netflow_expire()`, and `ng_netflow_flow_add()`.

Here is the call graph for this function:



7.70.2.16 `static int uma_ctor_flow (void * mem, int size, void * arg, int how)` [static]

Definition at line 120 of file `netflow.c`.

References `CACHESIZE`.

Referenced by `ng_netflow_cache_init()`.

7.70.2.17 `static void uma_dtor_flow (void * mem, int size, void * arg)` [static]

Definition at line 134 of file `netflow.c`.

Referenced by `ng_netflow_cache_init()`.

7.70.3 Variable Documentation

7.70.3.1 `const char rcs_id[]` [static]

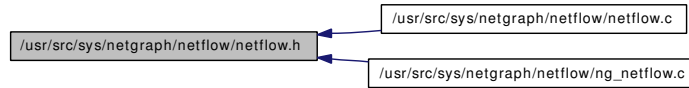
Initial value:

```
"@(#) $FreeBSD: src/sys/netgraph/netflow/netflow.c,v 1.24 2006/10/11 13:28:37 glebius Exp $"
```

Definition at line 30 of file `netflow.c`.

7.71 /usr/src/sys/netgraph/netflow/netflow.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [netflow_v1_header](#)
- struct [netflow_v5_header](#)
- struct [netflow_v1_record](#)
- struct [netflow_v5_record](#)
- struct [netflow_v5_export_dgram](#)

Defines

- #define [ACTIVE_TIMEOUT](#) (30*60)
- #define [INACTIVE_TIMEOUT](#) 15
- #define [NETFLOW_V1](#) 1
- #define [NETFLOW_V5](#) 5
- #define [NETFLOW_V1_MAX_RECORDS](#) 24
- #define [NETFLOW_V5_MAX_RECORDS](#) 30
- #define [NETFLOW_V1_MAX_SIZE](#)
- #define [NETFLOW_V5_MAX_SIZE](#)

Variables

- [netflow_v1_header __packed__](#)
- [netflow_v5_header __packed__](#)
- [netflow_v1_record __packed__](#)
- [netflow_v5_record __packed__](#)
- [netflow_v5_export_dgram __packed__](#)

7.71.1 Define Documentation

7.71.1.1 #define ACTIVE_TIMEOUT (30*60)

Definition at line 32 of file netflow.h.

Referenced by [ng_netflow_constructor\(\)](#).

7.71.1.2 #define INACTIVE_TIMEOUT 15

Definition at line 33 of file netflow.h.

Referenced by [ng_netflow_constructor\(\)](#).

7.71.1.3 #define NETFLOW_V1 1

Definition at line 47 of file netflow.h.

7.71.1.4 #define NETFLOW_V1_MAX_RECORDS 24

Definition at line 118 of file netflow.h.

7.71.1.5 #define NETFLOW_V1_MAX_SIZE

Value:

```
(sizeof(netflow_v1_header)+ \
    sizeof(netflow_v1_record)*NETFLOW_V1_MAX_RECORDS)
```

Definition at line 121 of file netflow.h.

7.71.1.6 #define NETFLOW_V5 5

Definition at line 48 of file netflow.h.

Referenced by get_export_dgram().

7.71.1.7 #define NETFLOW_V5_MAX_RECORDS 30

Definition at line 119 of file netflow.h.

Referenced by export_add().

7.71.1.8 #define NETFLOW_V5_MAX_SIZE

Value:

```
(sizeof(netflow_v5_header)+ \
    sizeof(netflow_v5_record)*NETFLOW_V5_MAX_RECORDS)
```

Definition at line 123 of file netflow.h.

7.71.2 Variable Documentation

7.71.2.1 struct netflow_v5_export_dgram __packed__

7.71.2.2 struct netflow_v5_record __packed__

7.71.2.3 struct netflow_v1_record __packed__

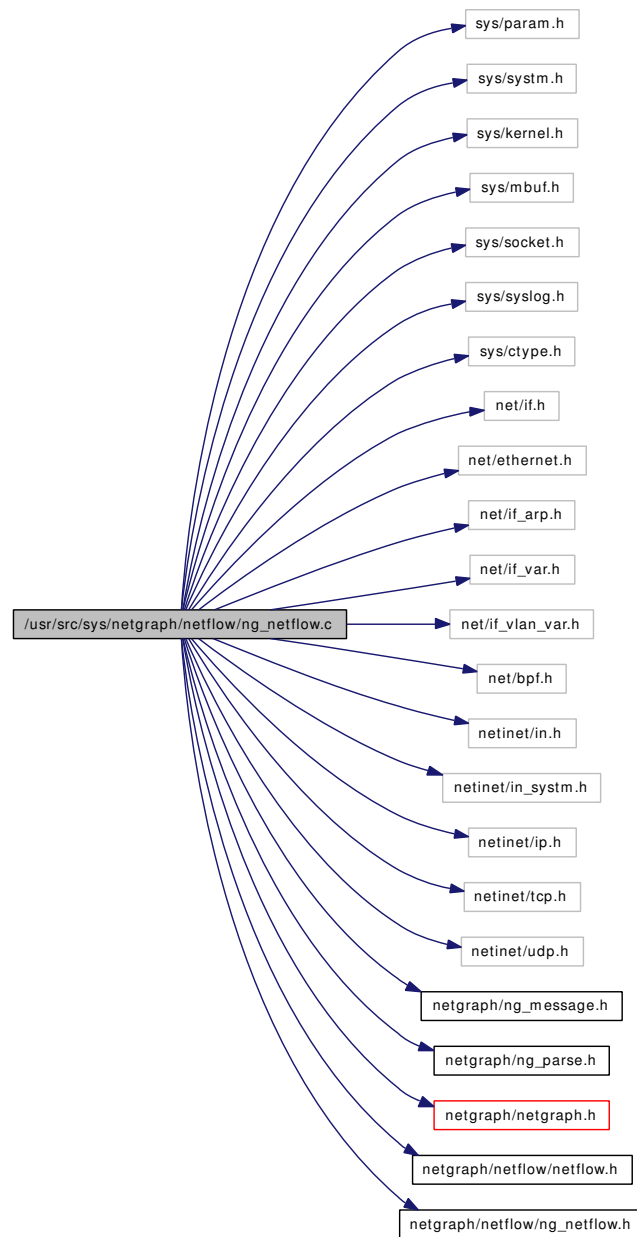
7.71.2.4 struct netflow_v5_header __packed__

7.71.2.5 struct netflow_v1_header __packed__

7.72 /usr/src/sys/netgraph/netflow/ng_netflow.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/ctype.h>
#include <net/if.h>
#include <net/ethernet.h>
#include <net/if_arp.h>
#include <net/if_var.h>
#include <net/if_vlan_var.h>
#include <net/bpf.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <netinet/udp.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/netgraph.h>
#include <netgraph/netflow/netflow.h>
#include <netgraph/netflow/ng_netflow.h>
```

Include dependency graph for ng_netflow.c:



Defines

- #define [M_CHECK](#)(length)

Functions

- [NETGRAPH_INIT](#) (netflow,&ng_netflow_tpestruct)
- static int [ng_netflow_constructor](#) (node_p node)
- static int [ng_netflow_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ng_netflow_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_netflow_rcvdata](#) (hook_p hook, item_p item)

- static int `ng_netflow_close` (`node_p` node)
- static int `ng_netflow_rmnode` (`node_p` node)
- static int `ng_netflow_disconnect` (`hook_p` hook)

Variables

- static const char `rsc_id` []
- static `ng_constructor_t` `ng_netflow_constructor`
- static `ng_rcvmsg_t` `ng_netflow_rcvmsg`
- static `ng_close_t` `ng_netflow_close`
- static `ng_shutdown_t` `ng_netflow_rmnode`
- static `ng_newhook_t` `ng_netflow_newhook`
- static `ng_rcvdata_t` `ng_netflow_rcvdata`
- static `ng_disconnect_t` `ng_netflow_disconnect`
- static struct `ng_parse_struct_field` `ng_netflow_info_type_fields` [] = NG_NETFLOW_INFO_TYPE
- static struct `ng_parse_type` `ng_netflow_info_type`
- static struct `ng_parse_struct_field` `ng_netflow_ifinfo_type_fields` [] = NG_NETFLOW_IFINFO_TYPE
- static struct `ng_parse_type` `ng_netflow_ifinfo_type`
- static struct `ng_parse_struct_field` `ng_netflow_setdlt_type_fields` [] = NG_NETFLOW_SETDLT_TYPE
- static struct `ng_parse_type` `ng_netflow_setdlt_type`
- static struct `ng_parse_struct_field` `ng_netflow_setifindex_type_fields` [] = NG_NETFLOW_SETIFINDEX_TYPE
- static struct `ng_parse_type` `ng_netflow_setifindex_type`
- static struct `ng_parse_struct_field` `ng_netflow_settimeouts_type_fields` [] = NG_NETFLOW_SETTIMEOUTS_TYPE
- static struct `ng_parse_type` `ng_netflow_settimeouts_type`
- static struct `ng_cmdlist` `ng_netflow_cmds` []
- static struct `ng_type` `ng_netflow_tpestruct`

7.72.1 Define Documentation

7.72.1.1 #define M_CHECK(length)

Value:

```
do {
    pullup_len += length;
    if ((m)->m_pkthdr.len < (pullup_len)) {
        error = EINVAL;
        goto bypass;
    }
    if ((m)->m_len < (pullup_len) &&
        ((m) = m_pullup((m), (pullup_len))) == NULL) {
        error = ENOBUFS;
        goto done;
    }
} while (0)
```

Referenced by `ng_netflow_rcvdata()`, and `ng_tcpmss_rcvdata()`.

7.72.2 Function Documentation

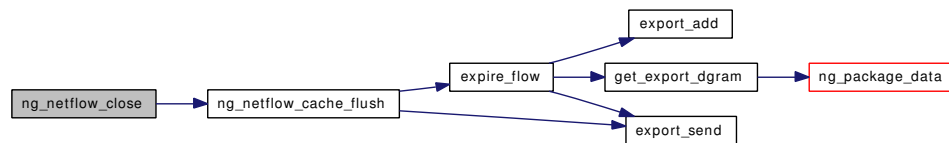
7.72.2.1 NETGRAPH_INIT ([netflow](#), & [ng_netflow_tpestruct](#))

7.72.2.2 `static int ng_netflow_close (node_p node)` [static]

Definition at line 613 of file `ng_netflow.c`.

References `ng_netflow_cache_flush()`, and `NG_NODE_PRIVATE`.

Here is the call graph for this function:



7.72.2.3 `static int ng_netflow_constructor (node_p node)` [static]

Definition at line 166 of file `ng_netflow.c`.

References `ACTIVE_TIMEOUT`, `INACTIVE_TIMEOUT`, `ng_netflow_cache_init()`, and `NG_NODE_SET_PRIVATE`.

Here is the call graph for this function:

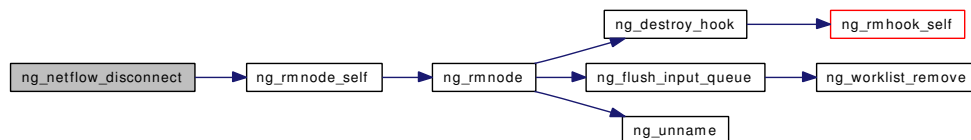


7.72.2.4 `static int ng_netflow_disconnect (hook_p hook)` [static]

Definition at line 639 of file `ng_netflow.c`.

References `ng_netflow_iface::hook`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, `ng_rmnode_self()`, and `ng_netflow_iface::out`.

Here is the call graph for this function:



7.72.2.5 `static int ng_netflow_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 201 of file `ng_netflow.c`.

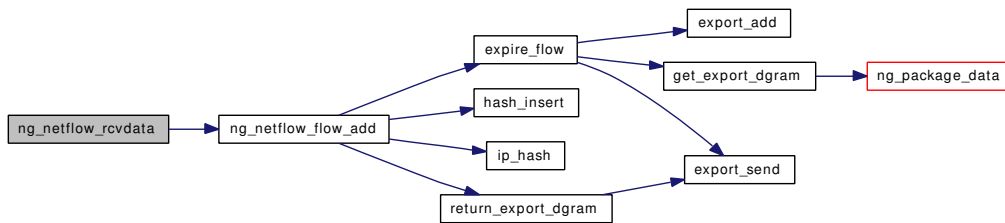
References `ng_netflow_iface::hook`, `ng_netflow_ifinfo::ifinfo_dlt`, `ng_netflow_iface::info`, `NG_HOOK_FORCE_QUEUE`, `NG_HOOK_PEER`, `NG_HOOK_SET_PRIVATE`, `ng_netflow_expire`, `NG_NETFLOW_HOOK_DATA`, `NG_NETFLOW_HOOK_EXPORT`, `NG_NETFLOW_HOOK_OUT`, `NG_NETFLOW_MAXIFACES`, `NG_NODE_PRIVATE`, and `ng_netflow_iface::out`.

7.72.2.6 `static int ng_netflow_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 442 of file `ng_netflow.c`.

References `ERROUT`, `ng_netflow_iface::hook`, `ng_netflow_ifinfo::ifinfo_dlt`, `ng_netflow_ifinfo::ifinfo_packets`, `ng_netflow_iface::info`, `M_CHECK`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_ITEM_HOOK`, `NG_FWD_NEW_DATA`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `ng_netflow_flow_add()`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `ng_netflow_iface::out`.

Here is the call graph for this function:

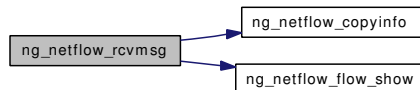


7.72.2.7 `static int ng_netflow_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 289 of file `ng_netflow.c`.

References `ng_netflow_settimeouts::active_timeout`, `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_netflow_setdlt::dlt`, `ERROUT`, `ng_mesg::header`, `ng_netflow_setifindex::iface`, `ng_netflow_setdlt::iface`, `ng_netflow_settimeouts::inactive_timeout`, `ng_netflow_setifindex::index`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `ng_netflow_copyinfo()`, `ng_netflow_flow_show()`, `NG_NETFLOW_MAXIFACES`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_NETFLOW_COOKIE`, `NGM_NETFLOW_IFINFO`, `NGM_NETFLOW_INFO`, `NGM_NETFLOW_SETDLT`, `NGM_NETFLOW_SETIFINDEX`, `NGM_NETFLOW_SETTIMEOUTS`, `NGM_NETFLOW_SHOW`, `NGRESP_SIZE`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.72.2.8 `static int ng_netflow_rmnode (node_p node)` [static]

Definition at line 625 of file `ng_netflow.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.72.3 Variable Documentation

7.72.3.1 `ng_close_t ng_netflow_close` [static]

Definition at line 62 of file ng_netflow.c.

7.72.3.2 `struct ng_cmdlist ng_netflow_cmds[]` [static]

Definition at line 109 of file ng_netflow.c.

7.72.3.3 `ng_constructor_t ng_netflow_constructor` [static]

Definition at line 60 of file ng_netflow.c.

7.72.3.4 `ng_disconnect_t ng_netflow_disconnect` [static]

Definition at line 66 of file ng_netflow.c.

7.72.3.5 `struct ng_parse_type ng_netflow_ifinfo_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_netflow_ifinfo_type_fields
}
```

Definition at line 79 of file ng_netflow.c.

7.72.3.6 `struct ng_parse_struct_field ng_netflow_ifinfo_type_fields[] = NG_NETFLOW_IFINFO_TYPE` [static]

Definition at line 78 of file ng_netflow.c.

7.72.3.7 `struct ng_parse_type ng_netflow_info_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_netflow_info_type_fields
}
```

Definition at line 71 of file ng_netflow.c.

7.72.3.8 `struct ng_parse_struct_field ng_netflow_info_type_fields[] = NG_NETFLOW_INFO_TYPE` [static]

Definition at line 70 of file ng_netflow.c.

7.72.3.9 `ng_newhook_t ng_netflow_newhook` [static]

Definition at line 64 of file ng_netflow.c.

7.72.3.10 `ng_rcvdata_t ng_netflow_rcvdata` [static]

Definition at line 65 of file ng_netflow.c.

7.72.3.11 `ng_rcvmsg_t ng_netflow_rcvmsg` [static]

Definition at line 61 of file ng_netflow.c.

7.72.3.12 `ng_shutdown_t ng_netflow_rmnode` [static]

Definition at line 63 of file ng_netflow.c.

7.72.3.13 `struct ng_parse_type ng_netflow_setdlt_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_netflow_setdlt_type_fields
}
```

Definition at line 87 of file ng_netflow.c.

7.72.3.14 `struct ng_parse_struct_field ng_netflow_setdlt_type_fields[] = NG_NETFLOW_SETDLT_TYPE` [static]

Definition at line 86 of file ng_netflow.c.

7.72.3.15 `struct ng_parse_type ng_netflow_setifindex_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_netflow_setifindex_type_fields
}
```

Definition at line 95 of file ng_netflow.c.

7.72.3.16 `struct ng_parse_struct_field ng_netflow_setifindex_type_fields[] = NG_NETFLOW_SETIFINDEX_TYPE` [static]

Definition at line 94 of file ng_netflow.c.

7.72.3.17 struct [ng_parse_type](#) [ng_netflow_settimeouts_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_netflow_settimeouts_type_fields
}
```

Definition at line 103 of file ng_netflow.c.

7.72.3.18 struct [ng_parse_struct_field](#) [ng_netflow_settimeouts_type_fields](#)[] = [NG_NETFLOW_SETTIMEOUTS_TYPE](#) [static]

Definition at line 102 of file ng_netflow.c.

7.72.3.19 struct [ng_type](#) [ng_netflow_typestruct](#) [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_NETFLOW_NODE_TYPE,
    .constructor =  ng_netflow_constructor,
    .rcvmsg =      ng_netflow_rcvmsg,
    .close =       ng_netflow_close,
    .shutdown =    ng_netflow_rmnode,
    .newhook =     ng_netflow_newhook,
    .rcvdata =     ng_netflow_rcvdata,
    .disconnect =  ng_netflow_disconnect,
    .cmdlist =     ng_netflow_cmds,
}
```

Definition at line 150 of file ng_netflow.c.

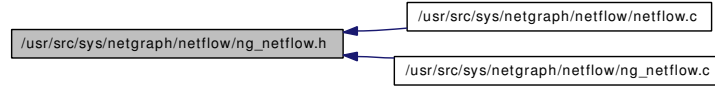
7.72.3.20 const char [rcs_id](#)[] [static]**Initial value:**

```
"@(#) $FreeBSD: src/sys/netgraph/netflow/ng_netflow.c,v 1.13 2006/10/11 15:27:13 glebius Exp $"
```

Definition at line 30 of file ng_netflow.c.

7.73 /usr/src/sys/netgraph/netflow/ng_netflow.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_netflow_info](#)
- struct [ng_netflow_ifinfo](#)
- struct [ng_netflow_setdl](#)
- struct [ng_netflow_setifindex](#)
- struct [ng_netflow_settimeouts](#)
- struct [flow_rec](#)
- struct [flow_entry_data](#)
- struct [ngnf_flows](#)
- struct [flow_entry](#)
- struct [ng_netflow_iface](#)
- struct [netflow](#)
- struct [flow_hash_entry](#)

Defines

- #define [NG_NETFLOW_NODE_TYPE](#) "netflow"
- #define [NGM_NETFLOW_COOKIE](#) 1137078102
- #define [NG_NETFLOW_MAXIFACES](#) 2048
- #define [NG_NETFLOW_HOOK_DATA](#) "iface"
- #define [NG_NETFLOW_HOOK_OUT](#) "out"
- #define [NG_NETFLOW_HOOK_EXPORT](#) "export"
- #define [MAXDLTNAMELEN](#) 20
- #define [r_ip_p](#) misc.i.prot
- #define [r_tos](#) misc.i.tos
- #define [r_i_ifx](#) misc.i.i_ifx
- #define [r_misc](#) misc.all
- #define [r_ports](#) ports.both
- #define [r_sport](#) ports.dir.s_port
- #define [r_dport](#) ports.dir.d_port
- #define [fle_i_ifx](#) r.misc.i.i_ifx
- #define [NREC_AT_ONCE](#) 1000
- #define [NGRESP_SIZE](#)
- #define [SORCVBUF_SIZE](#) (NGRESP_SIZE + 2 * sizeof(struct [ng_msg](#)))
- #define [NG_NETFLOW_INFO_TYPE](#)
- #define [NG_NETFLOW_IFINFO_TYPE](#)
- #define [NG_NETFLOW_SETDLT_TYPE](#)
- #define [NG_NETFLOW_SETIFINDEX_TYPE](#)
- #define [NG_NETFLOW_SETTIMEOUTS_TYPE](#)

- #define `CACHESIZE` (65536*4)
- #define `CACHELOWAT` (CACHESIZE * 3/4)
- #define `CACHEHIGHWAT` (CACHESIZE * 9/10)
- #define `ERROUT`(x) { error = (x); goto done; }

Typedefs

- typedef `ng_netflow_iface` * `iface_p`
- typedef `ng_netflow_ifinfo` * `ifinfo_p`
- typedef `netflow` * `priv_p`

Enumerations

- enum {
 `NGM_NETFLOW_INFO` = 1|`NGM_READONLY`|`NGM_HASREPLY`, `NGM_NETFLOW_IFINFO` = 2|`NGM_READONLY`|`NGM_HASREPLY`, `NGM_NETFLOW_SHOW` = 3|`NGM_READONLY`|`NGM_HASREPLY`, `NGM_NETFLOW_SETDLT` = 4,
 `NGM_NETFLOW_SETINDEX` = 5, `NGM_NETFLOW_SETTIMEOUTS` = 6 }

Functions

- int `ng_netflow_cache_init` (`priv_p`)
- void `ng_netflow_cache_flush` (`priv_p`)
- void `ng_netflow_copyinfo` (`priv_p`, struct `ng_netflow_info` *)
- int `ng_netflow_flow_add` (`priv_p`, struct ip *, `iface_p`, struct ifnet *)
- int `ng_netflow_flow_show` (`priv_p`, uint32_t last, struct `ng_mesg` *)

Variables

- timeout_t `ng_netflow_expire`

7.73.1 Define Documentation

7.73.1.1 #define `CACHEHIGHWAT` (CACHESIZE * 9/10)

Definition at line 234 of file `ng_netflow.h`.

7.73.1.2 #define `CACHELOWAT` (CACHESIZE * 3/4)

Definition at line 233 of file `ng_netflow.h`.

7.73.1.3 #define `CACHESIZE` (65536*4)

Definition at line 232 of file `ng_netflow.h`.

Referenced by `ng_netflow_cache_init`(), and `uma_ctor_flow`().

7.73.1.4 #define ERROUT(x) { error = (x); goto done; }

Definition at line 264 of file ng_netflow.h.

Referenced by ng_bpf_rcvmsg(), ng_deflate_rcvmsg(), ng_device_rcvmsg(), ng_ksocket_rcvmsg(), ng_mppc_rcvmsg(), ng_netflow_rcvdata(), ng_netflow_rcvmsg(), ng_ppp_rcvmsg(), ng_pptpgre_rcvmsg(), ng_pptpgre_rcv(), ng_pptpgre_xmit(), ng_pred1_rcvmsg(), ng_rfc1490_rcvdata(), ng_rfc1490_rcvmsg(), ng_tag_rcvmsg(), ng_tcpmss_rcvdata(), ng_tcpmss_rcvmsg(), ng_UI_rcvdata(), ng_vjc_rcvmsg(), nga_rcvmsg(), and ngt_rcvmsg().

7.73.1.5 #define fle_i_ifx r.misc.i.i_ifx

Definition at line 129 of file ng_netflow.h.

7.73.1.6 #define MAXDLTNAMELEN 20

Definition at line 72 of file ng_netflow.h.

7.73.1.7 #define NG_NETFLOW_HOOK_DATA "iface"

Definition at line 41 of file ng_netflow.h.

Referenced by ng_netflow_newhook().

7.73.1.8 #define NG_NETFLOW_HOOK_EXPORT "export"

Definition at line 43 of file ng_netflow.h.

Referenced by ng_netflow_newhook().

7.73.1.9 #define NG_NETFLOW_HOOK_OUT "out"

Definition at line 42 of file ng_netflow.h.

Referenced by ng_netflow_newhook().

7.73.1.10 #define NG_NETFLOW_IFINFO_TYPE

Value:

```
{
    { "packets",      &ng_parse_uint32_type },      \
    { "data link type", &ng_parse_uint8_type },    \
    { "index",      &ng_parse_uint16_type },      \
    { NULL }
}
```

Definition at line 181 of file ng_netflow.h.

7.73.1.11 #define NG_NETFLOW_INFO_TYPE**Value:**

```

{
    { "Bytes",          &ng_parse_uint64_type },          \
    { "Packets",       &ng_parse_uint32_type },          \
    { "Records used",  &ng_parse_uint32_type },          \
    { "Failed allocations", &ng_parse_uint32_type },    \
    { "Failed exports", &ng_parse_uint32_type },        \
    { "Active expiries", &ng_parse_uint32_type },        \
    { "Inactive expiries", &ng_parse_uint32_type },     \
    { "Inactive timeout", &ng_parse_uint32_type },     \
    { "Active timeout", &ng_parse_uint32_type },        \
    { NULL }
}

```

Definition at line 167 of file ng_netflow.h.

7.73.1.12 #define NG_NETFLOW_MAXIFACES 2048

Definition at line 37 of file ng_netflow.h.

Referenced by ng_netflow_newhook(), and ng_netflow_rcvmsg().

7.73.1.13 #define NG_NETFLOW_NODE_TYPE "netflow"

Definition at line 34 of file ng_netflow.h.

7.73.1.14 #define NG_NETFLOW_SETDLT_TYPE**Value:**

```

{
    { "iface",         &ng_parse_uint16_type },          \
    { "dlt",           &ng_parse_uint8_type },           \
    { NULL }
}

```

Definition at line 189 of file ng_netflow.h.

7.73.1.15 #define NG_NETFLOW_SETIFINDEX_TYPE**Value:**

```

{
    { "iface",         &ng_parse_uint16_type },          \
    { "index",         &ng_parse_uint16_type },          \
    { NULL }
}

```

Definition at line 196 of file ng_netflow.h.

7.73.1.16 #define NG_NETFLOW_SETTIMEOUTS_TYPE**Value:**

```
{
    { "inactive",    \
      &ng_parse_uint32_type }, \
    { "active",     \
      &ng_parse_uint32_type }, \
    { NULL }        \
}
```

Definition at line 203 of file ng_netflow.h.

7.73.1.17 #define NGM_NETFLOW_COOKIE 1137078102

Definition at line 35 of file ng_netflow.h.

Referenced by ng_netflow_rcvmsg().

7.73.1.18 #define NGRESP_SIZE**Value:**

```
(sizeof(struct ngnf_flows) + (NREC_AT_ONCE * \
                             sizeof(struct flow_entry_data)))
```

Definition at line 144 of file ng_netflow.h.

Referenced by ng_netflow_rcvmsg().

7.73.1.19 #define NREC_AT_ONCE 1000

Definition at line 143 of file ng_netflow.h.

Referenced by ng_netflow_flow_show().

7.73.1.20 #define r_dport ports.dir.d_port

Definition at line 122 of file ng_netflow.h.

7.73.1.21 #define r_i_ifx misc.i.i_ifx

Definition at line 118 of file ng_netflow.h.

7.73.1.22 #define r_ip_p misc.i.prot

Definition at line 116 of file ng_netflow.h.

7.73.1.23 #define r_misc misc.all

Definition at line 119 of file ng_netflow.h.

7.73.1.24 #define r_ports ports.both

Definition at line 120 of file ng_netflow.h.

7.73.1.25 #define r_sport ports.dir.s_port

Definition at line 121 of file ng_netflow.h.

7.73.1.26 #define r_tos misc.i.tos

Definition at line 117 of file ng_netflow.h.

7.73.1.27 #define SORCVBUF_SIZE (NGRESP_SIZE + 2 * sizeof(struct ng_mesg))

Definition at line 146 of file ng_netflow.h.

7.73.2 Typedef Documentation

7.73.2.1 typedef struct ng_netflow_iface* iface_p

Definition at line 216 of file ng_netflow.h.

7.73.2.2 typedef struct ng_netflow_ifinfo* ifinfo_p

Definition at line 217 of file ng_netflow.h.

7.73.2.3 typedef struct netflow* priv_p

Definition at line 256 of file ng_netflow.h.

7.73.3 Enumeration Type Documentation

7.73.3.1 anonymous enum

Enumerator:

NGM_NETFLOW_INFO
NGM_NETFLOW_IFINFO
NGM_NETFLOW_SHOW
NGM_NETFLOW_SETDLT
NGM_NETFLOW_SETIFINDEX
NGM_NETFLOW_SETTIMEOUTS

Definition at line 46 of file ng_netflow.h.

7.73.4 Function Documentation

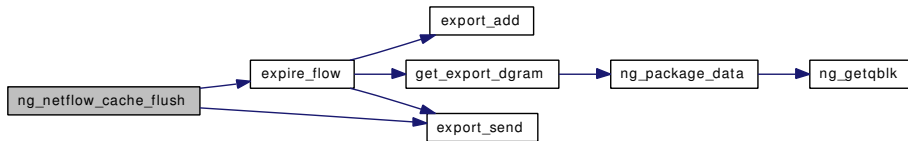
7.73.4.1 void ng_netflow_cache_flush (priv_p)

Definition at line 361 of file netflow.c.

References `expire_flow()`, `export_send()`, `flow_hash_entry::mtx`, `NBUCKETS`, and `NG_QUEUE`.

Referenced by `ng_netflow_close()`.

Here is the call graph for this function:



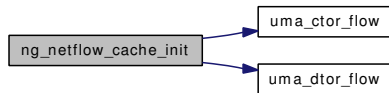
7.73.4.2 int ng_netflow_cache_init (priv_p)

Definition at line 328 of file netflow.c.

References `CACHESIZE`, `flow_hash_entry::mtx`, `NBUCKETS`, `uma_ctor_flow()`, and `uma_dtor_flow()`.

Referenced by `ng_netflow_constructor()`.

Here is the call graph for this function:



7.73.4.3 void ng_netflow_copyinfo (priv_p, struct ng_netflow_info *)

Definition at line 220 of file netflow.c.

Referenced by `ng_netflow_rcvmsg()`.

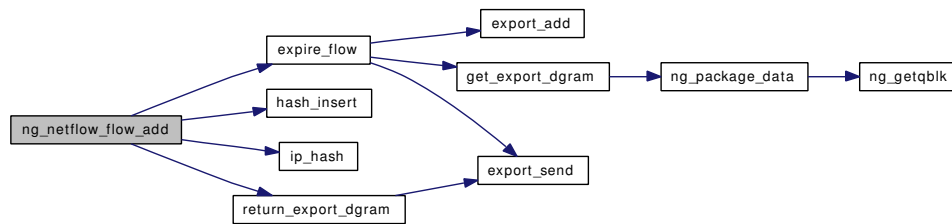
7.73.4.4 int ng_netflow_flow_add (priv_p, struct ip *, iface_p, struct ifnet *)

Definition at line 397 of file netflow.c.

References `AGED`, `flow_entry_data::bytes`, `expire_flow()`, `flow_entry::f`, `hash_insert()`, `ng_netflow_ifinfo::ifinfo_index`, `INACTIVE`, `ng_netflow_iface::info`, `ip_hash()`, `flow_entry_data::last`, `flow_hash_entry::mtx`, `NG_QUEUE`, `flow_entry_data::packets`, `flow_entry_data::r`, `return_export_dgram()`, `SMALL`, and `flow_entry_data::tcp_flags`.

Referenced by `ng_netflow_rcvdata()`.

Here is the call graph for this function:



7.73.4.5 int ng_netflow_flow_show (priv_p, uint32_t last, struct ng_mesg *)

Definition at line 533 of file netflow.c.

References `ng_mesg::data`, `ngnf_flows::entries`, `ngnf_flows::last`, `flow_hash_entry::mtx`, `NBUCKETS`, `ngnf_flows::nentries`, and `NREC_AT_ONCE`.

Referenced by `ng_netflow_rcvmsg()`.

7.73.5 Variable Documentation

7.73.5.1 timeout_t ng_netflow_expire

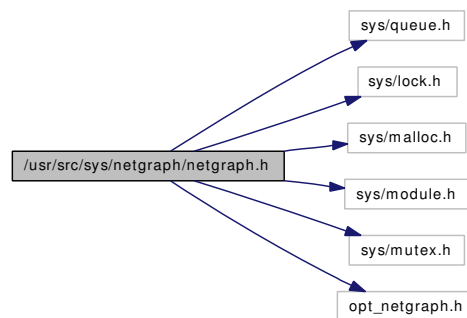
Definition at line 270 of file `ng_netflow.h`.

Referenced by `ng_netflow_expire()`, and `ng_netflow_newhook()`.

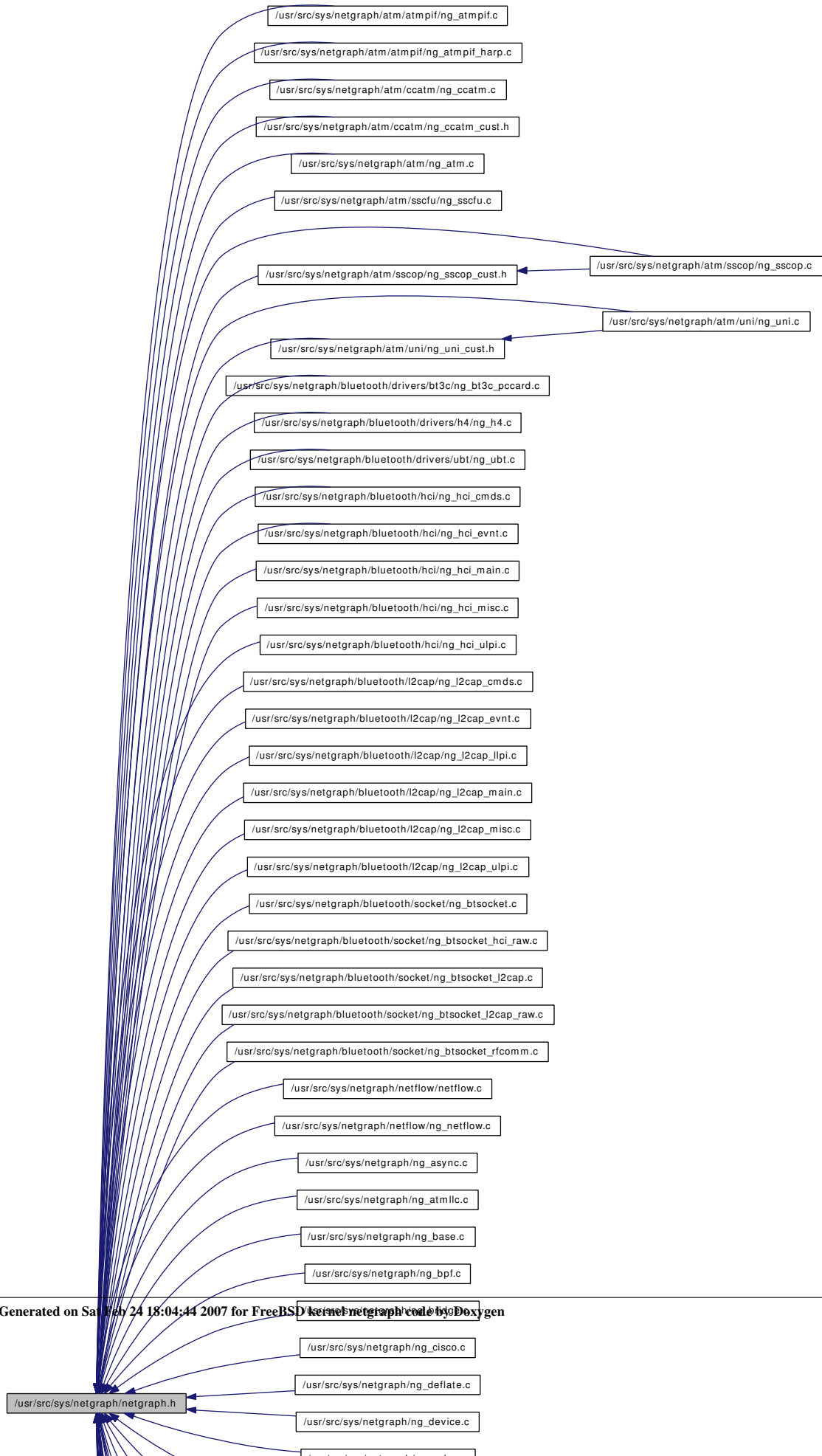
7.74 /usr/src/sys/netgraph/netgraph.h File Reference

```
#include <sys/queue.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/mutex.h>
#include "opt_netgraph.h"
```

Include dependency graph for netgraph.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_hook](#)
- struct [ng_queue](#)
- struct [ng_node](#)
- struct [ng_item](#)
- struct [ng_cmdlist](#)
- struct [ng_type](#)
- struct [ng_tag_prio](#)

Defines

- #define [NG_SEPARATE_MALLOC](#)
- #define [_NG_ABI_VERSION](#) 11
- #define [NG_ABI_VERSION](#) [_NG_ABI_VERSION](#)
- #define [HK_INVALID](#) 0x0001
- #define [HK_QUEUE](#) 0x0002
- #define [HK_FORCE_WRITER](#) 0x0004
- #define [HK_DEAD](#) 0x0008
- #define [_NG_HOOK_REF](#)(hook) atomic_add_int(&(hook) → hk_refs, 1)
- #define [_NG_HOOK_NAME](#)(hook) ((hook) → hk_name)
- #define [_NG_HOOK_UNREF](#)(hook) ng_unref_hook(hook)
- #define [_NG_HOOK_SET_PRIVATE](#)(hook, val) do {(hook) → hk_private = val;} while (0)
- #define [_NG_HOOK_SET_RCVMSG](#)(hook, val) do {(hook) → hk_rcvmsg = val;} while (0)
- #define [_NG_HOOK_SET_RCVDATA](#)(hook, val) do {(hook) → hk_rcvdata = val;} while (0)
- #define [_NG_HOOK_PRIVATE](#)(hook) ((hook) → hk_private)
- #define [_NG_HOOK_NOT_VALID](#)(hook) ((hook) → hk_flags & HK_INVALID)
- #define [_NG_HOOK_IS_VALID](#)(hook) (!(hook) → hk_flags & HK_INVALID))
- #define [_NG_HOOK_NODE](#)(hook) ((hook) → hk_node)
- #define [_NG_HOOK_PEER](#)(hook) ((hook) → hk_peer)
- #define [_NG_HOOK_FORCE_WRITER](#)(hook) do { hook → hk_flags |= HK_FORCE_WRITER; } while (0)
- #define [_NG_HOOK_FORCE_QUEUE](#)(hook) do { hook → hk_flags |= HK_QUEUE; } while (0)
- #define [NG_PEER_NODE](#)(hook) [NG_HOOK_NODE](#)([NG_HOOK_PEER](#)(hook))
- #define [NG_PEER_HOOK_NAME](#)(hook) [NG_HOOK_NAME](#)([NG_HOOK_PEER](#)(hook))
- #define [NG_PEER_NODE_NAME](#)(hook) [NG_NODE_NAME](#)([NG_PEER_NODE](#)(hook))
- #define [NG_HOOK_REF](#)(hook) [_NG_HOOK_REF](#)(hook)
- #define [NG_HOOK_NAME](#)(hook) [_NG_HOOK_NAME](#)(hook)
- #define [NG_HOOK_UNREF](#)(hook) [_NG_HOOK_UNREF](#)(hook)
- #define [NG_HOOK_SET_PRIVATE](#)(hook, val) [_NG_HOOK_SET_PRIVATE](#)(hook, val)
- #define [NG_HOOK_SET_RCVMSG](#)(hook, val) [_NG_HOOK_SET_RCVMSG](#)(hook, val)
- #define [NG_HOOK_SET_RCVDATA](#)(hook, val) [_NG_HOOK_SET_RCVDATA](#)(hook, val)
- #define [NG_HOOK_PRIVATE](#)(hook) [_NG_HOOK_PRIVATE](#)(hook)
- #define [NG_HOOK_NOT_VALID](#)(hook) [_NG_HOOK_NOT_VALID](#)(hook)
- #define [NG_HOOK_IS_VALID](#)(hook) [_NG_HOOK_IS_VALID](#)(hook)
- #define [NG_HOOK_NODE](#)(hook) [_NG_HOOK_NODE](#)(hook)
- #define [NG_HOOK_PEER](#)(hook) [_NG_HOOK_PEER](#)(hook)
- #define [NG_HOOK_FORCE_WRITER](#)(hook) [_NG_HOOK_FORCE_WRITER](#)(hook)
- #define [NG_HOOK_FORCE_QUEUE](#)(hook) [_NG_HOOK_FORCE_QUEUE](#)(hook)
- #define [NGF_INVALID](#) 0x00000001

- #define `NG_INVALID` `NGF_INVALID`
- #define `NGF_WORKQ` `0x00000002`
- #define `NG_WORKQ` `NGF_WORKQ`
- #define `NGF_FORCE_WRITER` `0x00000004`
- #define `NG_FORCE_WRITER` `NGF_FORCE_WRITER`
- #define `NGF_CLOSING` `0x00000008`
- #define `NG_CLOSING` `NGF_CLOSING`
- #define `NGF_REALLY_DIE` `0x00000010`
- #define `NG_REALLY_DIE` `NGF_REALLY_DIE`
- #define `NGF_TYPE1` `0x10000000`
- #define `NGF_TYPE2` `0x20000000`
- #define `NGF_TYPE3` `0x40000000`
- #define `NGF_TYPE4` `0x80000000`
- #define `_NG_NODE_NAME`(node) ((node) → nd_name + 0)
- #define `_NG_NODE_HAS_NAME`(node) ((node) → nd_name[0] + 0)
- #define `_NG_NODE_ID`(node) ((node) → nd_ID + 0)
- #define `_NG_NODE_REF`(node) `atomic_add_int(&(node) → nd_refs, 1)`
- #define `_NG_NODE_UNREF`(node) `ng_unref_node(node)`
- #define `_NG_NODE_SET_PRIVATE`(node, val) `do {(node) → nd_private = val;} while (0)`
- #define `_NG_NODE_PRIVATE`(node) ((node) → nd_private)
- #define `_NG_NODE_IS_VALID`(node) `!((node) → nd_flags & NGF_INVALID)`
- #define `_NG_NODE_NOT_VALID`(node) `((node) → nd_flags & NGF_INVALID)`
- #define `_NG_NODE_NUMHOOKS`(node) `((node) → nd_numhooks + 0)`
- #define `_NG_NODE_FORCE_WRITER`(node) `do{ node → nd_flags |= NGF_FORCE_WRITER; }while (0)`
- #define `_NG_NODE_REALLY_DIE`(node) `do{ node → nd_flags |= (NGF_REALLY_DIE|NGF_INVALID); }while (0)`
- #define `_NG_NODE_REVIVE`(node) `do { node → nd_flags &= ~NGF_INVALID; } while (0)`
- #define `_NG_NODE_FOREACH_HOOK`(node, fn, arg, rethook)
- #define `NG_NODE_NAME`(node) `_NG_NODE_NAME(node)`
- #define `NG_NODE_HAS_NAME`(node) `_NG_NODE_HAS_NAME(node)`
- #define `NG_NODE_ID`(node) `_NG_NODE_ID(node)`
- #define `NG_NODE_REF`(node) `_NG_NODE_REF(node)`
- #define `NG_NODE_UNREF`(node) `_NG_NODE_UNREF(node)`
- #define `NG_NODE_SET_PRIVATE`(node, val) `_NG_NODE_SET_PRIVATE(node, val)`
- #define `NG_NODE_PRIVATE`(node) `_NG_NODE_PRIVATE(node)`
- #define `NG_NODE_IS_VALID`(node) `_NG_NODE_IS_VALID(node)`
- #define `NG_NODE_NOT_VALID`(node) `_NG_NODE_NOT_VALID(node)`
- #define `NG_NODE_FORCE_WRITER`(node) `_NG_NODE_FORCE_WRITER(node)`
- #define `NG_NODE_REALLY_DIE`(node) `_NG_NODE_REALLY_DIE(node)`
- #define `NG_NODE_NUMHOOKS`(node) `_NG_NODE_NUMHOOKS(node)`
- #define `NG_NODE_REVIVE`(node) `_NG_NODE_REVIVE(node)`
- #define `NG_NODE_FOREACH_HOOK`(node, fn, arg, rethook) `_NG_NODE_FOREACH_HOOK(node, fn, arg, rethook)`
- #define `NGQF_TYPE` `0x03`
- #define `NGQF_MESG` `0x00`
- #define `NGQF_DATA` `0x01`
- #define `NGQF_FN` `0x02`
- #define `NGQF_UNDEF` `0x03`
- #define `NGQF_RW` `0x04`

- #define NGQF_READER 0x04
- #define NGQF_WRITER 0x00
- #define NGQF_QMODE 0x08
- #define NGQF_QREADER 0x08
- #define NGQF_QWRITER 0x00
- #define _NGI_M(i) ((i) → body.da_m)
- #define _NGI_MSG(i) ((i) → body.msg.msg_msg)
- #define _NGI_RETADDR(i) ((i) → body.msg.msg_retaddr)
- #define _NGI_FN(i) ((i) → body.fn.fn_fn)
- #define _NGI_ARG1(i) ((i) → body.fn.fn_arg1)
- #define _NGI_ARG2(i) ((i) → body.fn.fn_arg2)
- #define _NGI_NODE(i) ((i) → el_dest)
- #define _NGI_HOOK(i) ((i) → el_hook)
- #define _NGI_SET_HOOK(i, h) do { _NGI_HOOK(i) = h; h = NULL;} while (0)
- #define _NGI_CLR_HOOK(i)
- #define _NGI_SET_NODE(i, n) do { _NGI_NODE(i) = n; n = NULL;} while (0)
- #define _NGI_CLR_NODE(i)
- #define NGI_M(i) _NGI_M(i)
- #define NGI_MSG(i) _NGI_MSG(i)
- #define NGI_RETADDR(i) _NGI_RETADDR(i)
- #define NGI_FN(i) _NGI_FN(i)
- #define NGI_ARG1(i) _NGI_ARG1(i)
- #define NGI_ARG2(i) _NGI_ARG2(i)
- #define NGI_NODE(i) _NGI_NODE(i)
- #define NGI_HOOK(i) _NGI_HOOK(i)
- #define NGI_SET_HOOK(i, h) _NGI_SET_HOOK(i,h)
- #define NGI_CLR_HOOK(i) _NGI_CLR_HOOK(i)
- #define NGI_SET_NODE(i, n) _NGI_SET_NODE(i,n)
- #define NGI_CLR_NODE(i) _NGI_CLR_NODE(i)
- #define NG_FREE_ITEM(item) ng_free_item((item))
- #define SAVE_LINE(item) do { } while (0)
- #define NGI_GET_M(i, m)
- #define NGI_GET_MSG(i, m)
- #define NGI_GET_NODE(i, n)
- #define NGI_GET_HOOK(i, h)
- #define NGI_SET_WRITER(i) ((i) → el_flags &= ~NGQF_QMODE)
- #define NGI_SET_READER(i) ((i) → el_flags |= NGQF_QREADER)
- #define NGI_QUEUED_READER(i) ((i) → el_flags & NGQF_QREADER)
- #define NGI_QUEUED_WRITER(i) (((i) → el_flags & NGQF_QMODE) == NGQF_QWRITER)
- #define NG_FWD_ITEM_HOOK_FLAGS(error, item, hook, flags)
- #define NG_FWD_ITEM_HOOK(error, item, hook) NG_FWD_ITEM_HOOK_FLAGS(error, item, hook, NG_NOFLAGS)
- #define NG_FWD_NEW_DATA_FLAGS(error, item, hook, m, flags)
- #define NG_FWD_NEW_DATA(error, item, hook, m) NG_FWD_NEW_DATA_FLAGS(error, item, hook, m, NG_NOFLAGS)
- #define NG_SEND_DATA_FLAGS(error, hook, m, flags)
- #define NG_SEND_DATA_ONLY(error, hook, m) NG_SEND_DATA_FLAGS(error, hook, m, NG_NOFLAGS)
- #define NG_SEND_DATA(error, hook, m, x) NG_SEND_DATA_FLAGS(error, hook, m, NG_NOFLAGS)

- #define `NG_FREE_MSG`(msg)
- #define `NG_FREE_M`(m)
- #define `NG_SEND_MSG_HOOK`(error, here, msg, hook, retaddr)
- #define `NG_SEND_MSG_PATH`(error, here, msg, path, retaddr)
- #define `NG_SEND_MSG_ID`(error, here, msg, ID, retaddr)
- #define `NG_FWD_MSG_HOOK`(error, here, item, hook, retaddr)
- #define `NG_RESPOND_MSG`(error, here, item, resp)
- #define `NETGRAPH_INIT_ORDERED`(typename, typestructp, sub, order)
- #define `NETGRAPH_INIT`(tn, tp) `NETGRAPH_INIT_ORDERED`(tn, tp, `SI_SUB_PSEUDO`, `SI_ORDER_ANY`)
- #define `ng_send_fn`(node, hook, fn, arg1, arg2) `ng_send_fn1`(node, hook, fn, arg1, arg2, `NG_NOFLAGS`)
- #define `ng_callout_init`(c) `callout_init`(c, `NET_CALLOUT_MPSAFE`)
- #define `NG_NOFLAGS` 0x00000000
- #define `NG_QUEUE` 0x00000001
- #define `NG_WAITOK` 0x00000002
- #define `NG_PROGRESS` 0x00000004
- #define `NG_TAG_Prio` 1
- #define `NG_Prio_CUTOFF` 32
- #define `NG_Prio_LINKSTATE` 64
- #define `_NGI_META`(i) `NULL`
- #define `NGI_META`(i) `NULL`
- #define `NG_FREE_META`(meta)
- #define `NGI_GET_META`(i, m)
- #define `ng_copy_meta`(meta) `NULL`

Typedefs

- typedef `ng_item` * `item_p`
- typedef `ng_node` * `node_p`
- typedef `ng_hook` * `hook_p`
- typedef int `ng_constructor_t` (`node_p` node)
- typedef int `ng_close_t` (`node_p` node)
- typedef int `ng_shutdown_t` (`node_p` node)
- typedef int `ng_newhook_t` (`node_p` node, `hook_p` hook, const char *name)
- typedef `hook_p` `ng_findhook_t` (`node_p` node, const char *name)
- typedef int `ng_connect_t` (`hook_p` hook)
- typedef int `ng_rcvmsg_t` (`node_p` node, `item_p` item, `hook_p` lasthook)
- typedef int `ng_rcvdata_t` (`hook_p` hook, `item_p` item)
- typedef int `ng_disconnect_t` (`hook_p` hook)
- typedef int `ng_rcvitem` (`node_p` node, `hook_p` hook, `item_p` item)
- typedef int `ng_fn_eachhook` (`hook_p` hook, void *arg)
- typedef void `ng_item_fn` (`node_p` node, `hook_p` hook, void *arg1, int arg2)
- typedef void `ng_apply_t` (void *context, int error)
- typedef void * `meta_p`

Functions

- void `ng_unref_hook` (`hook_p` hook)
- int `ng_unref_node` (`node_p` node)
- `MALLOC_DECLARE` (M_NETGRAPH)
- `MALLOC_DECLARE` (M_NETGRAPH_MSG)
- int `ng_address_ID` (`node_p` here, `item_p` item, `ng_ID_t` ID, `ng_ID_t` retaddr)
- int `ng_address_hook` (`node_p` here, `item_p` item, `hook_p` hook, `ng_ID_t` retaddr)
- int `ng_address_path` (`node_p` here, `item_p` item, char *address, `ng_ID_t` raddr)
- int `ng_bypass` (`hook_p` hook1, `hook_p` hook2)
- `hook_p` `ng_findhook` (`node_p` node, const char *name)
- `ng_type` * `ng_findtype` (const char *type)
- int `ng_make_node_common` (struct `ng_type` *typep, `node_p` *nodep)
- int `ng_name_node` (`node_p` node, const char *name)
- int `ng_newtype` (struct `ng_type` *tp)
- `ng_ID_t` `ng_node2ID` (`node_p` node)
- `item_p` `ng_package_data` (struct mbuf *m, int flags)
- `item_p` `ng_package_msg` (struct `ng_mesg` *msg, int flags)
- `item_p` `ng_package_msg_self` (`node_p` here, `hook_p` hook, struct `ng_mesg` *msg)
- void `ng_replace_retaddr` (`node_p` here, `item_p` item, `ng_ID_t` retaddr)
- int `ng_rmhook_self` (`hook_p` hook)
- int `ng_rmnnode_self` (`node_p` here)
- int `ng_rmtime` (struct `ng_type` *tp)
- int `ng_snd_item` (`item_p` item, int queue)
- int `ng_send_fn1` (`node_p` node, `hook_p` hook, `ng_item_fn` *fn, void *arg1, int arg2, int flags)
- int `ng_uncallout` (struct callout *c, `node_p` node)
- int `ng_callout` (struct callout *c, `node_p` node, `hook_p` hook, int ticks, `ng_item_fn` *fn, void *arg1, int arg2)
- void `ng_free_item` (`item_p` item)
- int `ng_mod_event` (module_t mod, int what, void *arg)

7.74.1 Define Documentation

7.74.1.1 #define _NG_ABI_VERSION 11

Definition at line 69 of file netgraph.h.

7.74.1.2 #define _NG_HOOK_FORCE_QUEUE(hook) do { hook → hk_flags |= HK_QUEUE; } while (0)

Definition at line 149 of file netgraph.h.

7.74.1.3 #define _NG_HOOK_FORCE_WRITER(hook) do { hook → hk_flags |= HK_FORCE_WRITER; } while (0)

Definition at line 147 of file netgraph.h.

7.74.1.4 #define _NG_HOOK_IS_VALID(hook) (!(hook) → hk_flags & HK_INVALID))

Definition at line 144 of file netgraph.h.

7.74.1.5 #define _NG_HOOK_NAME(hook) ((hook) → hk_name)

Definition at line 137 of file netgraph.h.

7.74.1.6 #define _NG_HOOK_NODE(hook) ((hook) → hk_node)

Definition at line 145 of file netgraph.h.

Referenced by ng_destroy_hook(), and ng_unref_hook().

7.74.1.7 #define _NG_HOOK_NOT_VALID(hook) ((hook) → hk_flags & HK_INVALID)

Definition at line 143 of file netgraph.h.

7.74.1.8 #define _NG_HOOK_PEER(hook) ((hook) → hk_peer)

Definition at line 146 of file netgraph.h.

7.74.1.9 #define _NG_HOOK_PRIVATE(hook) ((hook) → hk_private)

Definition at line 142 of file netgraph.h.

7.74.1.10 #define _NG_HOOK_REF(hook) atomic_add_int(&(hook) → hk_refs, 1)

Definition at line 136 of file netgraph.h.

7.74.1.11 #define _NG_HOOK_SET_PRIVATE(hook, val) do {(hook) → hk_private = val;} while (0)

Definition at line 139 of file netgraph.h.

7.74.1.12 #define _NG_HOOK_SET_RCVDATA(hook, val) do {(hook) → hk_rcvdata = val;} while (0)

Definition at line 141 of file netgraph.h.

7.74.1.13 #define _NG_HOOK_SET_RCVMSG(hook, val) do {(hook) → hk_rcvmsg = val;} while (0)

Definition at line 140 of file netgraph.h.

7.74.1.14 #define _NG_HOOK_UNREF(hook) ng_unref_hook(hook)

Definition at line 138 of file netgraph.h.

7.74.1.15 #define _NG_NODE_FORCE_WRITER(node) do{ node → nd_flags |= NGF_FORCE_WRITER; }while (0)

Definition at line 383 of file netgraph.h.

7.74.1.16 #define _NG_NODE_FOREACH_HOOK(node, fn, arg, rethook)

Value:

```
do {
    hook_p _hook;
    (rethook) = NULL;
    LIST_FOREACH(_hook, &((node)->nd_hooks), hk_hooks) {
        if ((fn)(_hook, arg) == 0) {
            (rethook) = _hook;
            break;
        }
    }
} while (0)
```

Definition at line 396 of file netgraph.h.

7.74.1.17 #define _NG_NODE_HAS_NAME(node) ((node) → nd_name[0] + 0)

Definition at line 374 of file netgraph.h.

7.74.1.18 #define _NG_NODE_ID(node) ((node) → nd_ID + 0)

Definition at line 375 of file netgraph.h.

7.74.1.19 #define _NG_NODE_IS_VALID(node) (!((node) → nd_flags & NGF_INVALID))

Definition at line 380 of file netgraph.h.

7.74.1.20 #define _NG_NODE_NAME(node) ((node) → nd_name + 0)

Definition at line 373 of file netgraph.h.

7.74.1.21 #define _NG_NODE_NOT_VALID(node) ((node) → nd_flags & NGF_INVALID)

Definition at line 381 of file netgraph.h.

7.74.1.22 #define _NG_NODE_NUMHOOKS(node) ((node) → nd_numhooks + 0)

Definition at line 382 of file netgraph.h.

7.74.1.23 #define _NG_NODE_PRIVATE(node) ((node) → nd_private)

Definition at line 379 of file netgraph.h.

7.74.1.24 `#define _NG_NODE_REALLY_DIE(node) do{ node → nd_flags |= (NGF_REALLY_DIE|NGF_INVALID); }while (0)`

Definition at line 385 of file netgraph.h.

7.74.1.25 `#define _NG_NODE_REF(node) atomic_add_int(&(node) → nd_refs, 1)`

Definition at line 376 of file netgraph.h.

7.74.1.26 `#define _NG_NODE_REVIVE(node) do { node → nd_flags &= ~NGF_INVALID; } while (0)`

Definition at line 387 of file netgraph.h.

7.74.1.27 `#define _NG_NODE_SET_PRIVATE(node, val) do {(node) → nd_private = val;} while (0)`

Definition at line 378 of file netgraph.h.

7.74.1.28 `#define _NG_NODE_UNREF(node) ng_unref_node(node)`

Definition at line 377 of file netgraph.h.

Referenced by `ng_unref_hook()`.

7.74.1.29 `#define _NGI_ARG1(i) ((i) → body.fn.fn_arg1)`

Definition at line 641 of file netgraph.h.

Referenced by `ng_free_item()`.

7.74.1.30 `#define _NGI_ARG2(i) ((i) → body.fn.fn_arg2)`

Definition at line 642 of file netgraph.h.

Referenced by `ng_free_item()`.

7.74.1.31 `#define _NGI_CLR_HOOK(i)`

Value:

```
do {
    hook_p _hook = _NGI_HOOK(i);
    if (_hook) {
        _NG_HOOK_UNREF(_hook);
        _NGI_HOOK(i) = NULL;
    }
} while (0)
```

Definition at line 646 of file netgraph.h.

Referenced by `ng_free_item()`.

7.74.1.32 #define _NGI_CLR_NODE(i)**Value:**

```
do {
    node_p _node = _NGI_NODE(i);
    if (_node) {
        _NG_NODE_UNREF(_node);
        _NGI_NODE(i) = NULL;
    }
} while (0)
```

Definition at line 654 of file netgraph.h.

Referenced by ng_free_item().

7.74.1.33 #define _NGI_FN(i) ((i) → body.fn.fn_fn)

Definition at line 640 of file netgraph.h.

Referenced by ng_free_item().

7.74.1.34 #define _NGI_HOOK(i) ((i) → el_hook)

Definition at line 644 of file netgraph.h.

7.74.1.35 #define _NGI_M(i) ((i) → body.da_m)

Definition at line 637 of file netgraph.h.

Referenced by ng_free_item().

7.74.1.36 #define _NGI_META(i) NULL

Definition at line 1133 of file netgraph.h.

7.74.1.37 #define _NGI_MSG(i) ((i) → body.msg.msg_msg)

Definition at line 638 of file netgraph.h.

Referenced by ng_free_item(), and ng_hci_lp_acl_con_req().

7.74.1.38 #define _NGI_NODE(i) ((i) → el_dest)

Definition at line 643 of file netgraph.h.

7.74.1.39 #define _NGI_RETADDR(i) ((i) → body.msg.msg_retaddr)

Definition at line 639 of file netgraph.h.

Referenced by ng_free_item().

7.74.1.40 #define _NGI_SET_HOOK(i, h) do { _NGI_HOOK(i) = h; h = NULL;} while (0)

Definition at line 645 of file netgraph.h.

7.74.1.41 #define _NGI_SET_NODE(i, n) do { _NGI_NODE(i) = n; n = NULL;} while (0)

Definition at line 653 of file netgraph.h.

7.74.1.42 #define HK_DEAD 0x0008

Definition at line 129 of file netgraph.h.

7.74.1.43 #define HK_FORCE_WRITER 0x0004

Definition at line 128 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.44 #define HK_INVALID 0x0001

Definition at line 126 of file netgraph.h.

Referenced by ng_add_hook(), ng_con_nodes(), ng_con_part2(), ng_con_part3(), ng_destroy_hook(), and ng_mkpeer().

7.74.1.45 #define HK_QUEUE 0x0002

Definition at line 127 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.46 #define NETGRAPH_INIT(tn, tp) NETGRAPH_INIT_ORDERED(tn, tp, SI_SUB_PSEUDO, SI_ORDER_ANY)

Definition at line 1056 of file netgraph.h.

7.74.1.47 #define NETGRAPH_INIT_ORDERED(typename, typestructp, sub, order)**Value:**

```
static moduledata_t ng_##typename##_mod = {
    "ng_" #typename,
    ng_mod_event,
    (typestructp)
};
DECLARE_MODULE(ng_##typename, ng_##typename##_mod, sub, order);
MODULE_DEPEND(ng_##typename, netgraph, NG_ABI_VERSION,
              NG_ABI_VERSION,
              NG_ABI_VERSION)
```

Definition at line 1045 of file netgraph.h.

7.74.1.48 #define NG_ABI_VERSION _NG_ABI_VERSION

Definition at line 73 of file netgraph.h.

Referenced by ng_newtype().

7.74.1.49 #define ng_callout_init(c) callout_init(c, NET_CALLOUT_MPSAFE)

Definition at line 1100 of file netgraph.h.

Referenced by cisco_constructor(), ng_bridge_constructor(), ng_h4_open(), ng_hci_constructor(), ng_hci_new_con(), ng_l2cap_new_cmd(), ng_l2cap_new_con(), ng_l2tp_seq_init(), ng_ppp_constructor(), ng_pppoe_rcvmsg(), ng_pptpgre_constructor(), ng_source_constructor(), nglmi_constructor(), and ngt_open().

7.74.1.50 #define NG_CLOSING NGF_CLOSING

Definition at line 360 of file netgraph.h.

7.74.1.51 #define ng_copy_meta(meta) NULL

Definition at line 1137 of file netgraph.h.

7.74.1.52 #define NG_FORCE_WRITER NGF_FORCE_WRITER

Definition at line 358 of file netgraph.h.

7.74.1.53 #define NG_FREE_ITEM(item) ng_free_item((item))

Definition at line 781 of file netgraph.h.

Referenced by cisco_input(), cisco_rcvdata(), ng_address_hook(), ng_apply_item(), ng_atm_rcvdata(), ng_atm_rcvdrop(), ng_atmllc_rcvdata(), ng_atmpif_rcvdata(), ng_bpf_rcvdata(), ng_bridge_rcvdata(), ng_bt3c_rcvdata(), ng_bt3c_rcvmsg(), ng_btsocket_hci_raw_input(), ng_btsocket_hci_raw_node_rcvdata(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node_rcvdata(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_node_rcvdata(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_ccatm_rcvdata(), ng_ccatm_rcvdump(), ng_ccatm_rcvmanage(), ng_ccatm_rcvuni(), ng_connect_data(), ng_deflate_rcvdata(), ng_device_rcvdata(), ng_eiface_rcvdata(), ng_etf_rcvdata(), ng_ether_rcvdata(), ng_flush_input_queue(), ng_gif_demux_rcvdata(), ng_gif_rcvdata(), ng_h4_rcvdata(), ng_hci_acl_rcvdata(), ng_hci_drv_rcvdata(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_raw_rcvdata(), ng_hci_sco_rcvdata(), ng_hub_rcvdata(), ng_iface_rcvdata(), ng_ipfw_rcvdata(), ng_ksocket_rcvdata(), ng_l2cap_lower_rcvmsg(), ng_l2cap_rcvdata(), ng_l2cap_upper_rcvmsg(), ng_l2tp_rcvdata(), ng_l2tp_recv_ctrl(), ng_l2tp_recv_data(), ng_l2tp_recv_lower(), ng_mppc_rcvdata(), ng_nat_rcvdata(), ng_netflow_rcvdata(), ng_one2many_rcvdata(), ng_ppp_bypass(), ng_ppp_comp_recv(), ng_ppp_comp_xmit(), ng_ppp_crypt_xmit(), ng_ppp_link_xmit(), ng_ppp_mp_recv(), ng_ppp_mp_xmit(), ng_ppp_rcvdata(), ng_ppp_rcvdata_atalk(), ng_ppp_rcvdata_bypass(), ng_ppp_rcvdata_compress(), ng_ppp_rcvdata_decompress(), ng_ppp_rcvdata_decrypt(), ng_ppp_rcvdata_encrypt(), ng_ppp_rcvdata_inet(), ng_ppp_rcvdata_ipv6(), ng_ppp_rcvdata_ipx(), ng_ppp_rcvdata_vjc_comp(), ng_ppp_rcvdata_vjc_ip(), ng_ppp_rcvdata_vjc_uncomp(), ng_ppp_rcvdata_vjc_vjip(), ng_pppoe_rcvdata(), ng_pptpgre_rcvdata(), ng_pptpgre_recv(), ng_pptpgre_xmit(),

ng_pred1_rcvdata(), ng_rfc1490_rcvdata(), ng_snd_item(), ng_source_rcvdata(), ng_split_rcvdata(), ng_sppp_rcvdata(), ng_sscfu_rcvlower(), ng_sscfu_rcvupper(), ng_sscop_rcvlower(), ng_sscop_rcvmanage(), ng_sscop_rcvupper(), ng_tag_rcvdata(), ng_tcpmss_rcvdata(), ng_ubt_rcvdata(), ng_ubt_rcvmsg(), ng_UI_rcvdata(), ng_UI_rcvmsg(), ng_unicast_rcvlower(), ng_unicast_rcvupper(), ng_vjc_rcvdata(), ng_vlan_rcvdata(), ng_XXX_rcvdata(), nga_rcv_sync(), ngfrm_decode(), ngfrm_rcvdata(), ngh_rcvdata(), ngipi_rcvdata(), nglmi_rcvdata(), ngs_rcvdata(), ngs_rcvmsg(), ngt_rcvdata(), and send_data_packets().

7.74.1.54 #define NG_FREE_M(m)

Value:

```
do {
    if ((m) {
        m_freem((m));
        (m) = NULL;
    }
} while (0)
```

Definition at line 880 of file netgraph.h.

Referenced by bt3c_forward(), bt3c_pccard_detach(), bt3c_receive(), con_compl(), con_req(), data_buffer_overflow(), discon_compl(), encryption_change(), hardware_error(), inquiry_result(), mode_change(), ng_atm_rcvdata(), ng_bridge_rcvdata(), ng_bt3c_rcvdata(), ng_btsocket_hci_raw_data_input(), ng_btsocket_hci_raw_output(), ng_btsocket_hci_raw_send(), ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_raw_send(), ng_btsocket_l2cap_send(), ng_btsocket_rfcomm_prepare_packet(), ng_btsocket_rfcomm_receive_fc(), ng_btsocket_rfcomm_receive_frame(), ng_btsocket_rfcomm_receive_mcc(), ng_btsocket_rfcomm_receive_msc(), ng_btsocket_rfcomm_receive_pn(), ng_btsocket_rfcomm_receive_rls(), ng_btsocket_rfcomm_receive_rpn(), ng_btsocket_rfcomm_receive_test(), ng_btsocket_rfcomm_receive_uih(), ng_btsocket_rfcomm_send(), ng_btsocket_rfcomm_send_uih(), ng_deflate_compress(), ng_deflate_decompress(), ng_device_rcvdata(), ng_eiface_rcvdata(), ng_etc_rcvdata(), ng_ether_rcv_lower(), ng_ether_rcv_upper(), ng_free_item(), ng_gif_demux_rcvdata(), ng_gif_rcv_lower(), ng_h4_rcvdata(), ng_hci_acl_rcvdata(), ng_hci_process_command_complete(), ng_hci_process_command_status(), ng_hci_process_command_timeout(), ng_hci_process_event(), ng_hci_raw_rcvdata(), ng_hci_sco_rcvdata(), ng_iface_rcvdata(), ng_ipfw_rcvdata(), ng_l2cap_free_con(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_req(), ng_l2cap_lp_receive(), ng_l2cap_lp_send(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_con_req(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_echo_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_req(), ng_l2cap_process_info_rsp(), ng_l2cap_process_signal_cmd(), ng_l2cap_rcvdata(), ng_l2cap_receive(), ng_l2tp_rcv_data(), ng_l2tp_rcv_lower(), ng_mppc_rcvdata(), ng_netflow_rcvdata(), ng_one2many_rcvdata(), ng_package_data(), ng_ppp_frag_checkstale(), ng_ppp_frag_process(), ng_ppp_frag_reset(), ng_ppp_frag_trim(), ng_ppp_link_xmit(), ng_ppp_mp_rcv(), ng_ppp_mp_xmit(), ng_pppoe_rcvdata(), ng_pptpgre_rcv(), ng_pptpgre_xmit(), ng_pred1_compress(), ng_pred1_decompress(), ng_rfc1490_rcvdata(), ng_source_clr_data(), ng_source_rcvdata(), ng_sppp_rcvdata(), ng_tcpmss_rcvdata(), ng_ubt_rcvdata(), ng_UI_rcvdata(), ng_vjc_rcvdata(), ng_vlan_rcvdata(), ng_XXX_rcvdata(), nga_rcv_sync(), ngfrm_decode(), ngfrm_rcvdata(), nglmi_checkdata(), nglmi_rcvdata(), ngs_rcvdata(), num_compl_pkts(), page_scan_mode_change(), page_scan_rep_mode_change(), process_hc_baseband_params(), process_info_params(), process_link_control_params(), process_link_control_status(), process_link_policy_params(), process_link_policy_status(), process_status_params(), process_testing_params(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), read_remote_features_compl(), role_change(), send_l2cap_cfg_rsp(), ubt_bulk_in_complete2(), ubt_bulk_in_start(), ubt_bulk_out_start(), ubt_intr_complete2(), ubt_intr_start(), ubt_isoc_out_start(), and ubt_request_start().

7.74.1.55 #define NG_FREE_META(meta)

Definition at line 1135 of file netgraph.h.

7.74.1.56 #define NG_FREE_MSG(msg)**Value:**

```
do {
    if ((msg)) {
        FREE(msg, M_NETGRAPH_MSG);
        (msg) = NULL;
    }
} while (0)
```

Definition at line 872 of file netgraph.h.

Referenced by cisco_rcvmsg(), ng_atmpif_rcvmsg(), ng_bpf_rcvmsg(), ng_bt3c_rcvmsg(), ng_btsocket_hci_raw_control(), ng_btsocket_hci_raw_msg_input(), ng_btsocket_hci_raw_send_sync_ngmsg(), ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_send_sync_ngmsg(), ng_ccatm_rcvmsg(), ng_deflate_rcvmsg(), ng_device_rcvmsg(), ng_eiface_rcvmsg(), ng_etf_rcvmsg(), ng_ether_rcvmsg(), ng_fec_rcvmsg(), ng_free_item(), ng_generic_msg(), ng_gif_demux_rcvmsg(), ng_gif_rcvmsg(), ng_h4_rcvmsg(), ng_hci_default_rcvmsg(), ng_hci_lp_acl_con_req(), ng_iface_rcvmsg(), ng_ksocket_rcvmsg(), ng_l2cap_default_rcvmsg(), ng_l2cap_l2ca_enable_clt(), ng_l2tp_rcvmsg(), ng_mppc_rcvmsg(), ng_nat_rcvmsg(), ng_netflow_rcvmsg(), ng_one2many_rcvmsg(), ng_package_msg(), ng_package_msg_self(), ng_ppp_rcvmsg(), ng_pppoe_rcvmsg(), ng_pptpgre_rcvmsg(), ng_predl_rcvmsg(), ng_rfc1490_rcvmsg(), ng_source_rcvmsg(), ng_sppp_rcvmsg(), ng_sscfu_rcvmsg(), ng_sscop_rcvmsg(), ng_tag_rcvmsg(), ng_tcpmss_rcvmsg(), ng_ubt_rcvmsg(), ng_vjc_rcvmsg(), ng_xxx_rcvmsg(), nga_rcvmsg(), ngh_rcvmsg(), ngs_rcvmsg(), and ngt_rcvmsg().

7.74.1.57 #define NG_FWD_ITEM_HOOK(error, item, hook) NG_FWD_ITEM_HOOK_FLAGS(error, item, hook, NG_NOFLAGS)

Definition at line 832 of file netgraph.h.

Referenced by ng_bpf_rcvdata(), ng_gif_demux_rcvdata(), ng_hci_drv_rcvdata(), ng_hci_lp_acl_con_req(), ng_hub_rcvdata(), ng_nat_rcvdata(), ng_netflow_rcvdata(), ng_one2many_rcvdata(), ng_ppp_bypass(), ng_ppp_comp_recv(), ng_ppp_comp_xmit(), ng_ppp_crypt_recv(), ng_ppp_crypt_xmit(), ng_ppp_hcomp_recv(), ng_ppp_hcomp_xmit(), ng_ppp_proto_recv(), ng_ppp_rcvmsg(), ng_pppoe_rcvdata(), ng_split_rcvdata(), ng_tag_rcvdata(), ng_UI_rcvmsg(), nga_rcv_async(), nga_rcv_sync(), nge_rcvdata(), ngt_rcvdata(), ngt_rcvmsg(), and send_data_packets().

7.74.1.58 #define NG_FWD_ITEM_HOOK_FLAGS(error, item, hook, flags)**Value:**

```
do {
    (error) =
        ng_address_hook(NULL, (item), (hook), NG_NOFLAGS);
    if (error == 0) {
        SAVE_LINE(item);
        (error) = ng_snd_item((item), (flags));
    }
    (item) = NULL;
} while (0)
```

Definition at line 822 of file netgraph.h.

Referenced by export_send().

7.74.1.59 #define NG_FWD_MSG_HOOK(error, here, item, hook, retaddr)

Value:

```
do {
    if (((error) = ng_address_hook((here), (item),
                                   (hook), (retaddr))) == 0) {
        SAVE_LINE(item);
        (error) = ng_snd_item((item), 0);
    }
    (item) = NULL;
} while (0)
```

Definition at line 945 of file netgraph.h.

7.74.1.60 #define NG_FWD_NEW_DATA(error, item, hook, m) NG_FWD_NEW_DATA_FLAGS(error, item, hook, m, NG_NOFLAGS)

Definition at line 848 of file netgraph.h.

Referenced by cisco_input(), cisco_rcvdata(), ng_atmllc_rcvdata(), ng_bridge_rcvdata(), ng_deflate_rcvdata(), ng_etf_rcvdata(), ng_l2tp_rcv_data(), ng_l2tp_rcv_lower(), ng_mppc_rcvdata(), ng_netflow_rcvdata(), ng_ppp_link_xmit(), ng_pppoe_rcvdata(), ng_pptpgre_rcv(), ng_pptpgre_xmit(), ng_pred1_rcvdata(), ng_rfc1490_rcvdata(), ng_tcpmss_rcvdata(), ng_UI_rcvdata(), ng_vjc_rcvdata(), ng_vlan_rcvdata(), ng_xxx_rcvdata(), nga_rcv_async(), nga_rcv_sync(), ngfrm_decode(), and ngfrm_rcvdata().

7.74.1.61 #define NG_FWD_NEW_DATA_FLAGS(error, item, hook, m, flags)

Value:

```
do {
    NGI_M(item) = (m);
    (m) = NULL;
    NG_FWD_ITEM_HOOK_FLAGS(error, item, hook, flags);
} while (0)
```

Definition at line 842 of file netgraph.h.

7.74.1.62 #define NG_HOOK_FORCE_QUEUE(hook) _NG_HOOK_FORCE_QUEUE(hook)

Definition at line 309 of file netgraph.h.

Referenced by ng_atm_connect(), ng_bt3c_connect(), ng_btsocket_l2cap_node_connect(), ng_ccatm_newhook(), ng_gif_connect(), ng_h4_connect(), ng_ipfw_connect(), ng_ksocket_newhook(), ng_netflow_newhook(), ng_ubt_connect(), ng_xxx_connect(), and ngt_connect().

7.74.1.63 #define NG_HOOK_FORCE_WRITER(hook) _NG_HOOK_FORCE_WRITER(hook)

Definition at line 308 of file netgraph.h.

Referenced by `ng_ppp_newhook()`, and `nga_newhook()`.

7.74.1.64 #define NG_HOOK_IS_VALID(hook) _NG_HOOK_IS_VALID(hook)

Definition at line 305 of file `netgraph.h`.

Referenced by `bt3c_forward()`, `ng_findhook()`, `ng_h4_input()`, `ng_hci_lp_acl_con_req()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_ind()`, `ng_hci_mtap()`, `ng_ipfw_findhook1()`, `ng_l2cap_l2ca_enable_clt()`, and `send_data_packets()`.

7.74.1.65 #define NG_HOOK_NAME(hook) _NG_HOOK_NAME(hook)

Definition at line 298 of file `netgraph.h`.

Referenced by `ng_add_hook()`, `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `ng_atm_disconnect()`, `ng_atm_rcvmsg()`, `ng_bridge_rcvdata()`, `ng_btsocket_l2cap_default_msg_input()`, `ng_btsocket_l2cap_raw_input()`, `ng_ccatm_newhook()`, `ng_con_nodes()`, `ng_con_part2()`, `ng_findhook()`, `ng_generic_msg()`, `ng_hci_node_is_up()`, `ng_l2cap_send_hook_info()`, `ng_pppoe_rcvmsg()`, `ng_uni_disconnect()`, `ng_vlan_rcvmsg()`, `ngs_rcvdata()`, `pppoe_send_event()`, and `text_status()`.

7.74.1.66 #define NG_HOOK_NODE(hook) _NG_HOOK_NODE(hook)

Definition at line 306 of file `netgraph.h`.

Referenced by `cisco_disconnect()`, `cisco_rcvdata()`, `ng_apply_item()`, `ng_atm_disconnect()`, `ng_atm_rcvdata()`, `ng_atmllc_disconnect()`, `ng_atmllc_rcvdata()`, `ng_atmpif_disconnect()`, `ng_atmpif_rcvdata()`, `ng_bpf_disconnect()`, `ng_bridge_disconnect()`, `ng_bridge_rcvdata()`, `ng_bt3c_connect()`, `ng_bt3c_disconnect()`, `ng_bt3c_rcvdata()`, `ng_ccatm_disconnect()`, `ng_con_part3()`, `ng_deflate_disconnect()`, `ng_deflate_rcvdata()`, `ng_destroy_hook()`, `ng_device_disconnect()`, `ng_device_rcvdata()`, `ng_eiface_disconnect()`, `ng_eiface_rcvdata()`, `ng_etf_disconnect()`, `ng_etf_rcvdata()`, `ng_ether_disconnect()`, `ng_ether_rcvdata()`, `ng_gif_demux_disconnect()`, `ng_gif_demux_rcvdata()`, `ng_gif_disconnect()`, `ng_gif_rcvdata()`, `ng_h4_connect()`, `ng_h4_disconnect()`, `ng_h4_rcvdata()`, `ng_hci_acl_rcvdata()`, `ng_hci_connect()`, `ng_hci_disconnect()`, `ng_hci_drv_rcvdata()`, `ng_hci_raw_rcvdata()`, `ng_hci_sco_rcvdata()`, `ng_hub_disconnect()`, `ng_hub_rcvdata()`, `ng_iface_disconnect()`, `ng_iface_rcvdata()`, `ng_ksocket_connect()`, `ng_ksocket_disconnect()`, `ng_ksocket_rcvdata()`, `ng_l2cap_connect()`, `ng_l2cap_disconnect()`, `ng_l2cap_rcvdata()`, `ng_l2tp_disconnect()`, `ng_l2tp_rcvdata()`, `ng_mppc_disconnect()`, `ng_mppc_rcvdata()`, `ng_nat_disconnect()`, `ng_nat_rcvdata()`, `ng_netflow_disconnect()`, `ng_netflow_rcvdata()`, `ng_one2many_disconnect()`, `ng_one2many_rcvdata()`, `ng_ppp_disconnect()`, `ng_ppp_rcvdata()`, `ng_ppp_rcvdata_atalk()`, `ng_ppp_rcvdata_bypass()`, `ng_ppp_rcvdata_compress()`, `ng_ppp_rcvdata_decompress()`, `ng_ppp_rcvdata_decrypt()`, `ng_ppp_rcvdata_encrypt()`, `ng_ppp_rcvdata_inet()`, `ng_ppp_rcvdata_ip6()`, `ng_ppp_rcvdata_ipx()`, `ng_ppp_rcvdata_vjc_comp()`, `ng_ppp_rcvdata_vjc_ip()`, `ng_ppp_rcvdata_vjc_uncomp()`, `ng_ppp_rcvdata_vjc_vjip()`, `ng_pppoe_connect()`, `ng_pppoe_disconnect()`, `ng_pppoe_rcvdata()`, `ng_pppoe_sendpacket()`, `ng_pptp_disconnect()`, `ng_pptp_rcvdata()`, `ng_pred1_disconnect()`, `ng_pred1_rcvdata()`, `ng_rfc1490_disconnect()`, `ng_rfc1490_rcvdata()`, `ng_rmhook_self()`, `ng_snd_item()`, `ng_source_connect()`, `ng_source_disconnect()`, `ng_source_rcvdata()`, `ng_split_disconnect()`, `ng_split_rcvdata()`, `ng_sppp_disconnect()`, `ng_sppp_rcvdata()`, `ng_sscfu_disconnect()`, `ng_sscfu_rcvlower()`, `ng_sscfu_rcvupper()`, `ng_sscop_disconnect()`, `ng_sscop_rcvlower()`, `ng_sscop_rcvmanage()`, `ng_sscop_rcvupper()`, `ng_tag_disconnect()`, `ng_tag_setdata_in()`, `ng_tcpmss_disconnect()`, `ng_ubt_connect()`, `ng_ubt_disconnect()`, `ng_ubt_rcvdata()`, `ng_UI_disconnect()`, `ng_UI_rcvdata()`, `ng_uni_disconnect()`, `ng_uni_rcvlower()`, `ng_uni_rcvupper()`, `ng_vjc_disconnect()`, `ng_vjc_rcvdata()`, `ng_vlan_disconnect()`, `ng_vlan_rcvdata()`, `ng_xxx_disconnect()`, `ng_xxx_rcvdata()`, `nga_disconnect()`, `nga_rcvdata()`, `nge_disconnect()`, `ngfrm_disconnect()`, `ngfrm_rcvdata()`, `nggh_disconnect()`, `ngipi_disconnect()`, `nglmi_checkdata()`, `nglmi_`

disconnect(), nglmi_rcvdata(), ngs_connect(), ngs_disconnect(), ngs_rcvdata(), ng_t_disconnect(), ng_t_rcvdata(), pppoe_send_event(), pppoe_ticker(), send_acname(), send_sessionid(), and text_status().

7.74.1.67 #define NG_HOOK_NOT_VALID(hook) _NG_HOOK_NOT_VALID(hook)

Definition at line 304 of file netgraph.h.

Referenced by ng_address_hook(), ng_apply_item(), ng_btsocket_hci_raw_output(), ng_btsocket_l2cap_connect(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_raw_bind(), ng_btsocket_l2cap_raw_connect(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_rtclean(), ng_btsocket_l2cap_rtclean(), ng_btsocket_l2cap_send(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_l2cap_send_l2ca_cfg_rsp(), ng_btsocket_l2cap_send_l2ca_con_req(), ng_btsocket_l2cap_send_l2ca_con_rsp_req(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_generic_msg(), ng_hci_drv_rcvdata(), ng_hci_node_is_up(), ng_hci_send_command(), ng_l2cap_l2ca_cfg_ind(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_rsp(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_con_ind(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_con_rsp_rsp(), ng_l2cap_l2ca_discon_ind(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_lp_qos_req(), ng_l2cap_process_discon_timeout(), ng_l2cap_send_hook_info(), ng_path2noderef(), ng_snd_item(), sync_con_queue(), ubt_bulk_in_complete2(), ubt_intr_complete2(), and ubt_isoc_in_complete2().

7.74.1.68 #define NG_HOOK_PEER(hook) _NG_HOOK_PEER(hook)

Definition at line 307 of file netgraph.h.

Referenced by ng_address_hook(), ng_atm_connect(), ng_bt3c_connect(), ng_btsocket_l2cap_node_connect(), ng_destroy_hook(), ng_gif_connect(), ng_h4_connect(), ng_netflow_newhook(), ng_path2noderef(), ng_ubt_connect(), ng_xxx_connect(), ng_t_connect(), and text_status().

7.74.1.69 #define NG_HOOK_PRIVATE(hook) _NG_HOOK_PRIVATE(hook)

Definition at line 303 of file netgraph.h.

Referenced by cisco_disconnect(), cisco_rcvdata(), get_new_sid(), ng_atm_connect(), ng_atm_disconnect(), ng_atm_rcvdata(), ng_bpf_disconnect(), ng_bpf_rcvdata(), ng_bpf_rcvmsg(), ng_bpf_setprog(), ng_bridge_disconnect(), ng_bridge_rcvdata(), ng_btsocket_l2cap_data_input(), ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_l2ca_msg_input(), ng_btsocket_l2cap_node_disconnect(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_node_disconnect(), ng_ccatm_disconnect(), ng_ccatm_rcvdata(), ng_ccatm_rcvuni(), ng_etf_rcvdata(), ng_ipfw_disconnect(), ng_ipfw_findhook1(), ng_l2tp_disconnect(), ng_l2tp_find_session(), ng_l2tp_rcvdata(), ng_l2tp_rcvmsg(), ng_l2tp_rcv_lower(), ng_l2tp_reset_session(), ng_netflow_disconnect(), ng_netflow_rcvdata(), ng_one2many_disconnect(), ng_one2many_rcvdata(), ng_one2many_rcvmsg(), ng_ppp_disconnect(), ng_ppp_rcvdata(), ng_pppoe_disconnect(), ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), ng_split_disconnect(), ng_tag_disconnect(), ng_tag_rcvdata(), ng_tag_rcvmsg(), ng_tag_setdata_in(), ng_tag_setdata_out(), ng_tcpmss_disconnect(), ng_tcpmss_rcvdata(), ng_tcpmss_rcvmsg(), ng_vlan_disconnect(), ng_vlan_rcvdata(), ng_vlan_rcvmsg(), ng_xxx_connect(), ng_xxx_disconnect(), ng_xxx_rcvdata(), ngfrm_disconnect(), ngfrm_rcvdata(), ngh_disconnect(), ngh_rcvdata(), ngh_rcvmsg(), nglmi_disconnect(), nglmi_rcvdata(), ng_t_disconnect(), ng_t_rcvdata(), pppoe_broadcast_padi(), pppoe_find_svc(), pppoe_findsession(), pppoe_finduniq(), pppoe_match_svc(), and pppoe_ticker().

7.74.1.70 #define NG_HOOK_REF(hook) _NG_HOOK_REF(hook)

Definition at line 297 of file netgraph.h.

Referenced by ng_add_hook(), ng_address_hook(), ng_btsocket_l2cap_node_connect(), ng_btsocket_l2cap_node_rcvdata(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_connect(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_con_nodes(), ng_con_part2(), ng_mkpeer(), and ng_package_msg_self().

7.74.1.71 #define NG_HOOK_SET_PRIVATE(hook, val) _NG_HOOK_SET_PRIVATE(hook, val)

Definition at line 300 of file netgraph.h.

Referenced by cisco_newhook(), ng_atm_disconnect(), ng_atm_newhook(), ng_bpf_disconnect(), ng_bpf_newhook(), ng_bridge_newhook(), ng_btsocket_l2cap_default_msg_input(), ng_btsocket_l2cap_node_connect(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_node_connect(), ng_btsocket_l2cap_raw_rtclean(), ng_ccatm_disconnect(), ng_ccatm_newhook(), ng_eiface_newhook(), ng_etf_newhook(), ng_ipfw_disconnect(), ng_ipfw_newhook(), ng_l2tp_disconnect(), ng_l2tp_newhook(), ng_netflow_newhook(), ng_one2many_newhook(), ng_ppp_newhook(), ng_pppoe_disconnect(), ng_pppoe_newhook(), ng_split_newhook(), ng_sppp_newhook(), ng_tag_disconnect(), ng_tag_newhook(), ng_tcpmss_newhook(), ng_vlan_disconnect(), ng_vlan_newhook(), ng_vlan_rcvmsg(), ng_xxx_newhook(), ngfrm_newhook(), ngh_disconnect(), ngh_newhook(), nglimi_newhook(), ngs_newhook(), and ngt_newhook().

7.74.1.72 #define NG_HOOK_SET_RCVDATA(hook, val) _NG_HOOK_SET_RCVDATA(hook, val)

Definition at line 302 of file netgraph.h.

Referenced by ng_atm_newhook(), ng_ccatm_newhook(), ng_hci_connect(), ng_ppp_newhook(), ng_sscfu_newhook(), ng_sscop_newhook(), and ng_uni_newhook().

7.74.1.73 #define NG_HOOK_SET_RCVMSG(hook, val) _NG_HOOK_SET_RCVMSG(hook, val)

Definition at line 301 of file netgraph.h.

Referenced by ng_hci_connect(), and ng_l2cap_connect().

7.74.1.74 #define NG_HOOK_UNREF(hook) _NG_HOOK_UNREF(hook)

Definition at line 299 of file netgraph.h.

Referenced by ng_add_hook(), ng_apply_item(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_node_disconnect(), ng_btsocket_l2cap_raw_input(), ng_btsocket_l2cap_raw_node_disconnect(), ng_btsocket_l2cap_raw_rtclean(), ng_bypass(), ng_con_nodes(), ng_destroy_hook(), and ng_mkpeer().

7.74.1.75 #define NG_INVALID NGF_INVALID

Definition at line 354 of file netgraph.h.

7.74.1.76 #define NG_NODE_FORCE_WRITER(node) _NG_NODE_FORCE_WRITER(node)

Definition at line 568 of file netgraph.h.

Referenced by ng_bridge_constructor(), ng_deflate_constructor(), ng_h4_open(), ng_h4_shutdown(), ng_hci_constructor(), ng_l2cap_constructor(), ng_mppc_constructor(), ng_nat_constructor(), ng_pred1_constructor(), ng_sscop_constructor(), ng_ubt_shutdown(), and ng_uni_constructor().

7.74.1.77 #define NG_NODE_FOREACH_HOOK(node, fn, arg, rethook) _NG_NODE_FOREACH_HOOK(node, fn, arg, rethook)

Definition at line 572 of file netgraph.h.

Referenced by ng_l2tp_rcvmsg(), ng_l2tp_rcv_lower(), and ng_l2tp_seq_reset().

7.74.1.78 #define NG_NODE_HAS_NAME(node) _NG_NODE_HAS_NAME(node)

Definition at line 560 of file netgraph.h.

Referenced by ng_generic_msg(), ng_h4_ioctl(), ng_name2noderef(), ng_setsockaddr(), and ngt_tioctl().

7.74.1.79 #define NG_NODE_ID(node) _NG_NODE_ID(node)

Definition at line 561 of file netgraph.h.

Referenced by ng_atm_event_func(), ng_ksocket_finish_accept(), ng_ksocket_incoming2(), ng_ksocket_rcvdata(), ng_node2ID(), ng_pppoe_connect(), and ng_source_connect().

7.74.1.80 #define NG_NODE_IS_VALID(node) _NG_NODE_IS_VALID(node)

Definition at line 566 of file netgraph.h.

Referenced by cisco_disconnect(), ng_atmllc_disconnect(), ng_bpf_disconnect(), ng_bridge_disconnect(), ng_ccatm_disconnect(), ng_deflate_disconnect(), ng_ether_disconnect(), ng_generic_msg(), ng_gif_disconnect(), ng_hci_disconnect(), ng_hub_disconnect(), ng_ksocket_disconnect(), ng_l2cap_disconnect(), ng_l2tp_disconnect(), ng_mppc_disconnect(), ng_name2noderef(), ng_one2many_disconnect(), ng_ppp_disconnect(), ng_pppoe_disconnect(), ng_pptpgre_disconnect(), ng_pred1_disconnect(), ng_rfc1490_disconnect(), ng_rmnode(), ng_split_disconnect(), ng_ssfu_disconnect(), ng_sscop_disconnect(), ng_tag_disconnect(), ng_UI_disconnect(), ng_uni_disconnect(), ng_vjc_disconnect(), ng_vlan_disconnect(), ng_xxx_disconnect(), nga_disconnect(), nge_disconnect(), ngfrm_disconnect(), nglmi_disconnect(), ngs_disconnect(), and ngt_disconnect().

7.74.1.81 #define NG_NODE_NAME(node) _NG_NODE_NAME(node)

Definition at line 559 of file netgraph.h.

Referenced by complete_command(), data_buffer_overflow(), discon_compl(), encryption_change(), hardware_error(), mode_change(), ng_bridge_nodename(), ng_generic_msg(), ng_h4_input(), ng_h4_ioctl(), ng_h4_rcvdata(), ng_hci_acl_rcvdata(), ng_hci_command_timeout(), ng_hci_command_untimeout(), ng_hci_con_timeout(), ng_hci_con_untimeout(), ng_hci_drv_rcvdata(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_ind(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_ind(), ng_hci_lp_discon_req(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_ind(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_mtap(), ng_hci_node_is_up(), ng_hci_process_command_

complete(), ng_hci_process_command_timeout(), ng_hci_process_con_timeout(), ng_hci_process_event(), ng_hci_raw_rcvdata(), ng_hci_sco_rcvdata(), ng_hci_send_command(), ng_hci_send_data(), ng_l2cap_cmd_by_ident(), ng_l2cap_command_timeout(), ng_l2cap_command_untimeout(), ng_l2cap_con_fail(), ng_l2cap_con_ref(), ng_l2cap_con_unref(), ng_l2cap_con_wakeup(), ng_l2cap_discon_timeout(), ng_l2cap_discon_untimeout(), ng_l2cap_free_con(), ng_l2cap_l2ca_cfg_ind(), ng_l2cap_l2ca_cfg_req(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_req(), ng_l2cap_l2ca_cfg_rsp_rsp(), ng_l2cap_l2ca_clt_receive(), ng_l2cap_l2ca_con_ind(), ng_l2cap_l2ca_con_req(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_con_rsp_req(), ng_l2cap_l2ca_con_rsp_rsp(), ng_l2cap_l2ca_discon_ind(), ng_l2cap_l2ca_discon_req(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_req(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_req(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_receive(), ng_l2cap_l2ca_write_req(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lower_rcvmmsg(), ng_l2cap_lp_con_cfm(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_deliver(), ng_l2cap_lp_discon_ind(), ng_l2cap_lp_qos_cfm(), ng_l2cap_lp_qos_ind(), ng_l2cap_lp_qos_req(), ng_l2cap_lp_receive(), ng_l2cap_lp_send(), ng_l2cap_lp_timeout(), ng_l2cap_lp_untimeout(), ng_l2cap_new_cmd(), ng_l2cap_process_cfg_req(), ng_l2cap_process_cfg_rsp(), ng_l2cap_process_cmd_rej(), ng_l2cap_process_command_timeout(), ng_l2cap_process_con_rsp(), ng_l2cap_process_discon_req(), ng_l2cap_process_discon_rsp(), ng_l2cap_process_discon_timeout(), ng_l2cap_process_echo_req(), ng_l2cap_process_echo_rsp(), ng_l2cap_process_info_rsp(), ng_l2cap_process_lp_timeout(), ng_l2cap_process_signal_cmd(), ng_l2cap_receive(), ng_l2cap_send_hook_info(), ng_name2noderef(), ng_name_node(), ng_setsockaddr(), ngt_input(), ngt_tioctl(), num_compl_pkts(), process_link_policy_params(), qos_setup_compl(), qos_violation(), read_clock_offset_compl(), read_remote_features_compl(), role_change(), send_data_packets(), and text_status().

7.74.1.82 #define NG_NODE_NOT_VALID(node) _NG_NODE_NOT_VALID(node)

Definition at line 567 of file netgraph.h.

Referenced by ng_address_hook(), ng_apply_item(), ng_btsocket_hci_raw_output(), ng_generic_msg(), ng_h4_input(), ng_h4_start(), ng_hci_node_is_up(), ng_hci_process_command_timeout(), ng_hci_process_con_timeout(), ng_ksocket_incoming2(), ng_l2cap_process_command_timeout(), ng_l2cap_process_discon_timeout(), ng_l2cap_process_lp_timeout(), ng_l2cap_send_hook_info(), ng_path2noderef(), ng_ppp_frag_timeout(), and ng_snd_item().

7.74.1.83 #define NG_NODE_NUMHOOKS(node) _NG_NODE_NUMHOOKS(node)

Definition at line 570 of file netgraph.h.

Referenced by cisco_disconnect(), ng_atmllc_disconnect(), ng_bpf_disconnect(), ng_bridge_disconnect(), ng_ccatm_disconnect(), ng_deflate_disconnect(), ng_deflate_newhook(), ng_ether_disconnect(), ng_gif_disconnect(), ng_h4_ioctl(), ng_hci_disconnect(), ng_hub_disconnect(), ng_hub_rcvdata(), ng_ksocket_disconnect(), ng_l2cap_disconnect(), ng_l2tp_disconnect(), ng_mppc_disconnect(), ng_netflow_disconnect(), ng_one2many_disconnect(), ng_ppp_disconnect(), ng_pppoe_disconnect(), ng_pptpgre_disconnect(), ng_pred1_disconnect(), ng_pred1_newhook(), ng_rfc1490_disconnect(), ng_source_disconnect(), ng_split_disconnect(), ng_sscfu_disconnect(), ng_ssop_disconnect(), ng_tag_disconnect(), ng_tcpmss_disconnect(), ng_UI_disconnect(), ng_uni_disconnect(), ng_vjc_disconnect(), ng_vlan_disconnect(), ng_xxx_disconnect(), nga_disconnect(), ngd_send(), nge_disconnect(), ngfrm_disconnect(), ngh_disconnect(), ngipi_disconnect(), ngs_connect(), ngs_disconnect(), ngt_disconnect(), and ngt_tioctl().

7.74.1.84 #define NG_NODE_PRIVATE(node) _NG_NODE_PRIVATE(node)

Definition at line 565 of file netgraph.h.

Referenced by `bt3c_forward()`, `bt3c_send()`, `cisco_disconnect()`, `cisco_newhook()`, `cisco_rcvdata()`, `cisco_rcvmsg()`, `cisco_shutdown()`, `dump_saal_signal()`, `flow_lower()`, `flow_upper()`, `get_new_sid()`, `LMI_ticker()`, `ng_atm_cpcs_init()`, `ng_atm_cpcs_term()`, `ng_atm_detach()`, `ng_atm_disconnect()`, `ng_atm_event_func()`, `ng_atm_input()`, `ng_atm_input_orphans()`, `ng_atm_newhook()`, `ng_atm_output()`, `ng_atm_rcvdata()`, `ng_atm_rcvmsg()`, `ng_atm_shutdown()`, `ng_atmllc_disconnect()`, `ng_atmllc_newhook()`, `ng_atmllc_rcvdata()`, `ng_atmllc_shutdown()`, `ng_atmpif_disconnect()`, `ng_atmpif_newhook()`, `ng_atmpif_rcvdata()`, `ng_atmpif_rcvmsg()`, `ng_atmpif_rmnode()`, `ng_bridge_disconnect()`, `ng_bridge_newhook()`, `ng_bridge_rcvdata()`, `ng_bridge_rcvmsg()`, `ng_bridge_shutdown()`, `ng_bridge_timeout()`, `ng_bt3c_connect()`, `ng_bt3c_disconnect()`, `ng_bt3c_newhook()`, `ng_bt3c_rcvdata()`, `ng_bt3c_rcvmsg()`, `ng_bt3c_shutdown()`, `ng_ccatm_disconnect()`, `ng_ccatm_dump()`, `ng_ccatm_get_addresses()`, `ng_ccatm_newhook()`, `ng_ccatm_rcvmsg()`, `ng_ccatm_shutdown()`, `ng_connect_data()`, `ng_deflate_compress()`, `ng_deflate_decompress()`, `ng_deflate_disconnect()`, `ng_deflate_newhook()`, `ng_deflate_rcvdata()`, `ng_deflate_rcvmsg()`, `ng_deflate_reset_req()`, `ng_deflate_shutdown()`, `ng_device_disconnect()`, `ng_device_newhook()`, `ng_device_rcvdata()`, `ng_device_rcvmsg()`, `ng_eiface_disconnect()`, `ng_eiface_newhook()`, `ng_eiface_rcvdata()`, `ng_eiface_rcvmsg()`, `ng_eiface_rmnode()`, `ng_etf_disconnect()`, `ng_etf_newhook()`, `ng_etf_rcvdata()`, `ng_etf_rcvmsg()`, `ng_etf_shutdown()`, `ng_ether_detach()`, `ng_ether_disconnect()`, `ng_ether_input()`, `ng_ether_input_orphan()`, `ng_ether_link_state()`, `ng_ether_newhook()`, `ng_ether_output()`, `ng_ether_rcv_lower()`, `ng_ether_rcv_upper()`, `ng_ether_rcvdata()`, `ng_ether_rcvmsg()`, `ng_ether_shutdown()`, `ng_fec_input()`, `ng_fec_rcvmsg()`, `ng_fec_shutdown()`, `ng_gif_demux_disconnect()`, `ng_gif_demux_newhook()`, `ng_gif_demux_rcvdata()`, `ng_gif_demux_shutdown()`, `ng_gif_detach()`, `ng_gif_disconnect()`, `ng_gif_input()`, `ng_gif_input2()`, `ng_gif_input_orphan()`, `ng_gif_newhook()`, `ng_gif_rcv_lower()`, `ng_gif_rcvdata()`, `ng_gif_rcvmsg()`, `ng_gif_shutdown()`, `ng_h4_connect()`, `ng_h4_disconnect()`, `ng_h4_newhook()`, `ng_h4_process_timeout()`, `ng_h4_rcvdata()`, `ng_h4_rcvmsg()`, `ng_h4_shutdown()`, `ng_h4_start2()`, `ng_h4_timeout()`, `ng_h4_untimeout()`, `ng_hci_acl_rcvdata()`, `ng_hci_connect()`, `ng_hci_default_rcvmsg()`, `ng_hci_disconnect()`, `ng_hci_drv_rcvdata()`, `ng_hci_newhook()`, `ng_hci_node_is_up()`, `ng_hci_process_command_timeout()`, `ng_hci_process_con_timeout()`, `ng_hci_raw_rcvdata()`, `ng_hci_sco_rcvdata()`, `ng_hci_shutdown()`, `ng_hci_upper_rcvmsg()`, `ng_iface_disconnect()`, `ng_iface_newhook()`, `ng_iface_rcvdata()`, `ng_iface_rcvmsg()`, `ng_iface_shutdown()`, `ng_ksocket_connect()`, `ng_ksocket_finish_accept()`, `ng_ksocket_incoming2()`, `ng_ksocket_newhook()`, `ng_ksocket_rcvdata()`, `ng_ksocket_rcvmsg()`, `ng_ksocket_shutdown()`, `ng_l2cap_connect()`, `ng_l2cap_default_rcvmsg()`, `ng_l2cap_disconnect()`, `ng_l2cap_lower_rcvmsg()`, `ng_l2cap_newhook()`, `ng_l2cap_process_command_timeout()`, `ng_l2cap_process_discon_timeout()`, `ng_l2cap_process_lp_timeout()`, `ng_l2cap_rcvdata()`, `ng_l2cap_send_hook_info()`, `ng_l2cap_shutdown()`, `ng_l2cap_upper_rcvmsg()`, `ng_l2tp_disconnect()`, `ng_l2tp_newhook()`, `ng_l2tp_rcvdata()`, `ng_l2tp_rcvmsg()`, `ng_l2tp_recv_ctrl()`, `ng_l2tp_recv_data()`, `ng_l2tp_recv_lower()`, `ng_l2tp_seq_rack_timeout()`, `ng_l2tp_seq_xack_timeout()`, `ng_l2tp_shutdown()`, `ng_mppc_compress()`, `ng_mppc_decompress()`, `ng_mppc_disconnect()`, `ng_mppc_newhook()`, `ng_mppc_rcvdata()`, `ng_mppc_rcvmsg()`, `ng_mppc_reset_req()`, `ng_mppc_shutdown()`, `ng_nat_disconnect()`, `ng_nat_newhook()`, `ng_nat_rcvdata()`, `ng_nat_rcvmsg()`, `ng_nat_shutdown()`, `ng_netflow_close()`, `ng_netflow_disconnect()`, `ng_netflow_newhook()`, `ng_netflow_rcvdata()`, `ng_netflow_rcvmsg()`, `ng_netflow_rmnode()`, `ng_one2many_disconnect()`, `ng_one2many_newhook()`, `ng_one2many_rcvdata()`, `ng_one2many_rcvmsg()`, `ng_one2many_shutdown()`, `ng_ppp_bypass()`, `ng_ppp_check_packet()`, `ng_ppp_comp_rcv()`, `ng_ppp_comp_xmit()`, `ng_ppp_config_valid()`, `ng_ppp_crypt_rcv()`, `ng_ppp_crypt_xmit()`, `ng_ppp_disconnect()`, `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, `ng_ppp_frag_reset()`, `ng_ppp_frag_trim()`, `ng_ppp_get_packet()`, `ng_ppp_hcomp_rcv()`, `ng_ppp_hcomp_xmit()`, `ng_ppp_link_xmit()`, `ng_ppp_mp_rcv()`, `ng_ppp_mp_strategy()`, `ng_ppp_mp_xmit()`, `ng_ppp_newhook()`, `ng_ppp_proto_rcv()`, `ng_ppp_rcvdata()`, `ng_ppp_rcvdata_atalk()`, `ng_ppp_rcvdata_compress()`, `ng_ppp_rcvdata_decompress()`, `ng_ppp_rcvdata_decrypt()`, `ng_ppp_rcvdata_encrypt()`, `ng_ppp_rcvdata_inet()`, `ng_ppp_rcvdata_ipv6()`, `ng_ppp_rcvdata_ipx()`, `ng_ppp_rcvdata_vjc_comp()`, `ng_ppp_rcvdata_vjc_ip()`, `ng_ppp_rcvdata_vjc_uncomp()`, `ng_ppp_rcvdata_vjc_vjip()`, `ng_ppp_rcvmsg()`, `ng_ppp_shutdown()`, `ng_ppp_start_frag_timer()`, `ng_ppp_stop_frag_timer()`, `ng_ppp_update()`, `ng_pppoe_connect()`, `ng_pppoe_disconnect()`, `ng_pppoe_newhook()`, `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, `ng_pppoe_sendpacket()`, `ng_pppoe_shutdown()`, `ng_pptpgre_disconnect()`, `ng_pptpgre_newhook()`, `ng_pptpgre_rcvdata()`, `ng_pptpgre_rcvmsg()`, `ng_pptpgre_rcv()`, `ng_pptpgre_recv_ack_timeout()`, `ng_pptpgre_reset()`, `ng_pptpgre_send_ack_timeout()`, `ng_pptpgre_shutdown()`, `ng_`

pptpgre_start_recv_ack_timer(), ng_pptpgre_start_send_ack_timer(), ng_pptpgre_stop_recv_ack_timer(), ng_pptpgre_stop_send_ack_timer(), ng_pptpgre_time(), ng_pptpgre_xmit(), ng_pred1_compress(), ng_pred1_decompress(), ng_pred1_newhook(), ng_pred1_rcvdata(), ng_pred1_rcvmsg(), ng_pred1_shutdown(), ng_rfc1490_disconnect(), ng_rfc1490_newhook(), ng_rfc1490_rcvdata(), ng_rfc1490_rcvmsg(), ng_rfc1490_shutdown(), ng_source_connect(), ng_source_disconnect(), ng_source_newhook(), ng_source_rcvdata(), ng_source_rcvmsg(), ng_source_rmnode(), ng_split_newhook(), ng_split_rcvdata(), ng_split_shutdown(), ng_sppp_disconnect(), ng_sppp_newhook(), ng_sppp_rcvdata(), ng_sppp_rcvmsg(), ng_sppp_shutdown(), ng_sscfu_disconnect(), ng_sscfu_newhook(), ng_sscfu_rcvlower(), ng_sscfu_rcvmsg(), ng_sscfu_rcvupper(), ng_sscfu_shutdown(), ng_sscop_disconnect(), ng_sscop_newhook(), ng_sscop_rcvlower(), ng_sscop_rcvmanage(), ng_sscop_rcvmsg(), ng_sscop_rcvupper(), ng_sscop_shutdown(), ng_ubt_connect(), ng_ubt_disconnect(), ng_ubt_newhook(), ng_ubt_rcvdata(), ng_ubt_rcvmsg(), ng_ubt_shutdown(), ng_UI_disconnect(), ng_UI_newhook(), ng_UI_rcvdata(), ng_UI_rcvmsg(), ng_UI_shutdown(), ng_uni_disconnect(), ng_uni_newhook(), ng_uni_rcvlower(), ng_uni_rcvmsg(), ng_uni_rcvupper(), ng_uni_shutdown(), ng_vjc_disconnect(), ng_vjc_newhook(), ng_vjc_rcvdata(), ng_vjc_rcvmsg(), ng_vjc_shutdown(), ng_vlan_disconnect(), ng_vlan_newhook(), ng_vlan_rcvdata(), ng_vlan_rcvmsg(), ng_vlan_shutdown(), ng_xxx_newhook(), ng_xxx_rcvdata(), ng_xxx_rcvmsg(), ng_xxx_shutdown(), nga_disconnect(), nga_newhook(), nga_rcvdata(), nga_rcvmsg(), nga_shutdown(), ngc_send(), ngfrm_decode(), ngfrm_disconnect(), ngfrm_newhook(), ngfrm_rcvdata(), ngfrm_shutdown(), nglmi_checkdata(), nglmi_disconnect(), nglmi_newhook(), nglmi_rcvdata(), nglmi_rcvmsg(), nglmi_shutdown(), ngs_connect(), ngs_disconnect(), ngs_newhook(), ngs_rcvdata(), ngs_rcvmsg(), ngs_shutdown(), ngt_close(), ngt_disconnect(), ngt_newhook(), ngt_rcvdata(), ngt_rcvmsg(), ngt_shutdown(), ngt_timeout(), pppoe_broadcast_padi(), pppoe_find_svc(), pppoe_findsession(), pppoe_finduniq(), pppoe_match_svc(), pppoe_start(), pppoe_ticker(), Pred1Compress(), Pred1Decompress(), Pred1Init(), Pred1SyncTable(), sscfu_send_lower(), sscfu_send_upper(), sscop_send_lower(), sscop_send_manage(), sscop_send_upper(), text_status(), ubt_bulk_in_complete2(), ubt_bulk_out_complete2(), ubt_intr_complete2(), ubt_isoc_in_complete2(), ubt_isoc_out_complete2(), ubt_request_complete2(), uni_saal_output(), uni_uni_output(), vatmpif_harp_attach(), and vatmpif_harp_detach().

7.74.1.85 #define NG_NODE REALLY_DIE(node) _NG_NODE REALLY_DIE(node)

Definition at line 569 of file netgraph.h.

Referenced by ng_atm_detach(), ng_ether_detach(), and ng_gif_detach().

7.74.1.86 #define NG_NODE_REF(node) _NG_NODE_REF(node)

Definition at line 562 of file netgraph.h.

Referenced by ng_add_hook(), ng_address_hook(), ng_attach_cntl(), ng_con_part2(), ng_ID2noderef(), ng_make_node_common(), ng_name2noderef(), ng_package_msg_self(), ng_path2noderef(), ng_rmnode(), ng_setisr(), ubt_bulk_in_start(), ubt_bulk_out_start(), ubt_intr_start(), ubt_isoc_in_start(), ubt_isoc_out_start(), and ubt_request_start().

7.74.1.87 #define NG_NODE_REVIVE(node) _NG_NODE_REVIVE(node)

Definition at line 571 of file netgraph.h.

Referenced by ng_atm_shutdown(), ng_ether_shutdown(), ng_gif_shutdown(), and ng_xxx_shutdown().

7.74.1.88 #define NG_NODE_SET_PRIVATE(node, val) _NG_NODE_SET_PRIVATE(node, val)

Definition at line 564 of file netgraph.h.

Referenced by `bt3c_pccard_attach()`, `bt3c_pccard_detach()`, `cisco_constructor()`, `cisco_shutdown()`, `ng_atm_attach()`, `ng_atm_shutdown()`, `ng_atmllc_constructor()`, `ng_atmpif_constructor()`, `ng_atmpif_rmnode()`, `ng_attach_cntl()`, `ng_bpf_constructor()`, `ng_bridge_constructor()`, `ng_bridge_shutdown()`, `ng_bt3c_shutdown()`, `ng_ccatm_constructor()`, `ng_ccatm_shutdown()`, `ng_deflate_constructor()`, `ng_deflate_shutdown()`, `ng_device_constructor()`, `ng_eiface_constructor()`, `ng_eiface_rmnode()`, `ng_etf_constructor()`, `ng_etf_shutdown()`, `ng_ether_attach()`, `ng_ether_shutdown()`, `ng_fec_constructor()`, `ng_fec_shutdown()`, `ng_gif_attach()`, `ng_gif_demux_constructor()`, `ng_gif_demux_shutdown()`, `ng_gif_shutdown()`, `ng_h4_close()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_hci_constructor()`, `ng_hci_shutdown()`, `ng_iface_constructor()`, `ng_iface_shutdown()`, `ng_ksocket_constructor()`, `ng_ksocket_shutdown()`, `ng_l2cap_constructor()`, `ng_l2cap_shutdown()`, `ng_l2tp_constructor()`, `ng_mppc_constructor()`, `ng_mppc_shutdown()`, `ng_nat_constructor()`, `ng_nat_shutdown()`, `ng_netflow_constructor()`, `ng_netflow_rmnode()`, `ng_one2many_constructor()`, `ng_one2many_shutdown()`, `ng_ppp_constructor()`, `ng_ppp_shutdown()`, `ng_pppoe_constructor()`, `ng_pppoe_shutdown()`, `ng_pptpgre_constructor()`, `ng_pred1_constructor()`, `ng_pred1_shutdown()`, `ng_rfc1490_constructor()`, `ng_rfc1490_shutdown()`, `ng_source_constructor()`, `ng_source_rmnode()`, `ng_split_constructor()`, `ng_split_shutdown()`, `ng_sppp_constructor()`, `ng_sppp_shutdown()`, `ng_sscfu_constructor()`, `ng_sscfu_shutdown()`, `ng_sscop_constructor()`, `ng_sscop_shutdown()`, `ng_ubt_shutdown()`, `ng_UI_constructor()`, `ng_UI_shutdown()`, `ng_uni_constructor()`, `ng_uni_shutdown()`, `ng_vjc_constructor()`, `ng_vjc_shutdown()`, `ng_vlan_constructor()`, `ng_vlan_shutdown()`, `ng_xxx_constructor()`, `ng_xxx_shutdown()`, `nga_constructor()`, `nga_shutdown()`, `ngfrm_constructor()`, `ngfrm_shutdown()`, `nglmi_constructor()`, `nglmi_shutdown()`, `ngs_shutdown()`, `ngt_constructor()`, `ngt_open()`, `ngt_shutdown()`, and `USB_DETACH()`.

7.74.1.89 #define NG_NODE_UNREF(node) _NG_NODE_UNREF(node)

Definition at line 563 of file `netgraph.h`.

Referenced by `bt3c_pccard_attach()`, `cisco_shutdown()`, `ng_atm_attach()`, `ng_atm_shutdown()`, `ng_atmllc_shutdown()`, `ng_atmpif_rmnode()`, `ng_bpf_shutdown()`, `ng_bridge_shutdown()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_ccatm_shutdown()`, `ng_deflate_shutdown()`, `ng_destroy_hook()`, `ng_device_shutdown()`, `ng_eiface_rmnode()`, `ng_etf_shutdown()`, `ng_ether_attach()`, `ng_ether_shutdown()`, `ng_fec_shutdown()`, `ng_generic_msg()`, `ng_gif_attach()`, `ng_gif_demux_shutdown()`, `ng_gif_shutdown()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_hci_shutdown()`, `ng_iface_shutdown()`, `ng_ipfw_shutdown()`, `ng_ksocket_finish_accept()`, `ng_ksocket_shutdown()`, `ng_l2cap_shutdown()`, `ng_l2tp_shutdown()`, `ng_make_node()`, `ng_mppc_shutdown()`, `ng_name_node()`, `ng_nat_shutdown()`, `ng_netflow_rmnode()`, `ng_one2many_shutdown()`, `ng_path2noderef()`, `ng_ppp_shutdown()`, `ng_pppoe_shutdown()`, `ng_pptpgre_shutdown()`, `ng_pred1_shutdown()`, `ng_rfc1490_shutdown()`, `ng_rmnode()`, `ng_snd_item()`, `ng_socket_free_priv()`, `ng_source_rmnode()`, `ng_split_shutdown()`, `ng_sppp_shutdown()`, `ng_sscfu_shutdown()`, `ng_sscop_shutdown()`, `ng_tag_shutdown()`, `ng_ubt_shutdown()`, `ng_UI_shutdown()`, `ng_uni_shutdown()`, `ng_vjc_shutdown()`, `ng_vlan_shutdown()`, `ng_worklist_remove()`, `ng_xxx_shutdown()`, `nga_shutdown()`, `ngfrm_shutdown()`, `ngintr()`, `nglmi_shutdown()`, `ngs_shutdown()`, `ngt_open()`, `ngt_shutdown()`, `ubt_bulk_in_complete()`, `ubt_bulk_in_start()`, `ubt_bulk_out_complete()`, `ubt_bulk_out_start()`, `ubt_intr_complete()`, `ubt_intr_start()`, `ubt_isoc_in_complete()`, `ubt_isoc_in_start()`, `ubt_isoc_out_complete()`, `ubt_isoc_out_start()`, `ubt_request_complete()`, and `ubt_request_start()`.

7.74.1.90 #define NG_NOFLAGS 0x00000000

Definition at line 1103 of file `netgraph.h`.

Referenced by `get_export_dgram()`, `ng_netflow_expire()`, `ng_package_msg_self()`, `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, and `ng_ppp_mp_xmit()`.

**7.74.1.91 #define NG_PEER_HOOK_NAME(hook) NG_HOOK_NAME(NG_HOOK_-
PEER(hook))**

Definition at line 153 of file netgraph.h.

Referenced by ng_generic_msg().

7.74.1.92 #define NG_PEER_NODE(hook) NG_HOOK_NODE(NG_HOOK_PEER(hook))

Definition at line 152 of file netgraph.h.

Referenced by ng_address_hook(), ng_btsocket_hci_raw_output(), ng_generic_msg(), ng_path2noderef(), ng_pppoe_connect(), and ng_source_connect().

**7.74.1.93 #define NG_PEER_NODE_NAME(hook) NG_NODE_NAME(NG_PEER_-
NODE(hook))**

Definition at line 154 of file netgraph.h.

Referenced by ng_btsocket_hci_raw_node_rcvdata(), ng_btsocket_hci_raw_output(), and ng_generic_msg().

7.74.1.94 #define NG_PRIO_CUTOFF 32

Definition at line 1126 of file netgraph.h.

7.74.1.95 #define NG_PRIO_LINKSTATE 64

Definition at line 1127 of file netgraph.h.

Referenced by nglmi_inquire().

7.74.1.96 #define NG_PROGRESS 0x00000004

Definition at line 1106 of file netgraph.h.

Referenced by ng_snd_item(), and ngc_send().

7.74.1.97 #define NG_QUEUE 0x00000001

Definition at line 1104 of file netgraph.h.

Referenced by ng_ksocket_incoming(), ng_netflow_cache_flush(), ng_netflow_flow_add(), and ng_snd_item().

7.74.1.98 #define NG REALLY_DIE NGF REALLY_DIE

Definition at line 362 of file netgraph.h.

7.74.1.99 #define NG_RESPOND_MSG(error, here, item, resp)**Value:**

```

do {
    if (resp) {
        ng_ID_t _dest = NGI_RETADDR(item);
        NGI_RETADDR(item) = 0;
        NGI_MSG(item) = resp;
        if ((error = ng_address_ID((here), (item),
            _dest, 0)) == 0) {
            SAVE_LINE(item);
            (error) = ng_snd_item((item), NG_QUEUE);
        }
    } else
        NG_FREE_ITEM(item);
    (item) = NULL;
} while (0)

```

Definition at line 959 of file netgraph.h.

Referenced by `cisco_rcvmsg()`, `ng_atmlc_rcvmsg()`, `ng_atmpif_rcvmsg()`, `ng_bpf_rcvmsg()`, `ng_bt3c_rcvmsg()`, `ng_ccatm_rcvmsg()`, `ng_deflate_rcvmsg()`, `ng_device_rcvmsg()`, `ng_eiface_rcvmsg()`, `ng_etf_rcvmsg()`, `ng_ether_rcvmsg()`, `ng_fec_rcvmsg()`, `ng_gif_demux_rcvmsg()`, `ng_gif_rcvmsg()`, `ng_h4_rcvmsg()`, `ng_hci_default_rcvmsg()`, `ng_iface_rcvmsg()`, `ng_ksocket_rcvmsg()`, `ng_l2cap_default_rcvmsg()`, `ng_l2tp_rcvmsg()`, `ng_mppc_rcvmsg()`, `ng_nat_rcvmsg()`, `ng_netflow_rcvmsg()`, `ng_one2many_rcvmsg()`, `ng_ppp_rcvmsg()`, `ng_pppoe_rcvmsg()`, `ng_pptpgre_rcvmsg()`, `ng_pred1_rcvmsg()`, `ng_rfc1490_rcvmsg()`, `ng_source_rcvmsg()`, `ng_sppp_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_tag_rcvmsg()`, `ng_tcpmss_rcvmsg()`, `ng_ubt_rcvmsg()`, `ng_vjc_rcvmsg()`, `ng_xxx_rcvmsg()`, `nga_rcvmsg()`, `nge_rcvmsg()`, `ngh_rcvmsg()`, and `ngt_rcvmsg()`.

7.74.1.100 #define NG_SEND_DATA(error, hook, m, x) NG_SEND_DATA_FLAGS(error, hook, m, NG_NOFLAGS)

Definition at line 869 of file netgraph.h.

7.74.1.101 #define NG_SEND_DATA_FLAGS(error, hook, m, flags)**Value:**

```

do {
    item_p _item;
    if ((_item = ng_package_data(m, flags))) {
        NG_FWD_ITEM_HOOK_FLAGS(error, _item, hook, flags);
    } else {
        (error) = ENOMEM;
    }
    (m) = NULL;
} while (0)

```

Definition at line 855 of file netgraph.h.

Referenced by `ngd_send()`.

7.74.1.102 #define NG_SEND_DATA_ONLY(error, hook, m) NG_SEND_DATA_FLAGS(error, hook, m, NG_NOFLAGS)

Definition at line 866 of file netgraph.h.

Referenced by `bt3c_forward()`, `cisco_send()`, `ng_atm_input()`, `ng_atm_input_orphans()`, `ng_atm_output()`, `ng_atmpif_transmit()`, `ng_bridge_rcvdata()`, `ng_btsocket_hci_raw_output()`, `ng_btsocket_l2cap_send2()`, `ng_ccatm_dump()`, `ng_ccatm_respond_user()`, `ng_ccatm_send_uni()`, `ng_ccatm_send_uni_glob()`, `ng_ccatm_send_user()`, `ng_eiface_start2()`, `ng_ether_input()`, `ng_ether_input_orphan()`, `ng_ether_output()`, `ng_gif_input2()`, `ng_h4_input()`, `ng_hci_mtap()`, `ng_hci_send_command()`, `ng_hub_rcvdata()`, `ng_iface_send()`, `ng_ipfw_input()`, `ng_ksocket_incoming2()`, `ng_l2cap_l2ca_clt_receive()`, `ng_l2cap_l2ca_receive()`, `ng_l2cap_lp_deliver()`, `ng_l2tp_xmit_ctrl()`, `ng_one2many_rcvdata()`, `ng_pppoe_disconnect()`, `ng_pppoe_sendpacket()`, `ng_pptpgre_xmit()`, `ng_source_send()`, `ng_sppp_start()`, `nga_rcv_async()`, `ngdwrite()`, `nglmi_inquire()`, `ngt_input()`, `ngt_rcvdata()`, `pppoe_broadcast_padi()`, `pppoe_ticker()`, `sscfu_send_lower()`, `sscfu_send_upper()`, `sscop_send_lower()`, `sscop_send_manage()`, `sscop_send_upper()`, `ubt_bulk_in_complete2()`, `ubt_intr_complete2()`, `ubt_isoc_in_complete2()`, `uni_saal_output()`, and `uni_uni_output()`.

7.74.1.103 #define ng_send_fn(node, hook, fn, arg1, arg2) ng_send_fn1(node, hook, fn, arg1, arg2, NG_NOFLAGS)

Definition at line 1095 of file `netgraph.h`.

Referenced by `bt3c_swi_intr()`, `ng_atm_event()`, `ng_bt3c_rcvdata()`, `ng_btsocket_hci_raw_send()`, `ng_con_nodes()`, `ng_con_part2()`, `ng_eiface_start()`, `ng_h4_start()`, `ng_hci_connect()`, `ng_ksocket_connect()`, `ng_l2cap_connect()`, `ng_rmhook_self()`, `ng_rmnnode_self()`, `ubt_bulk_in_complete()`, `ubt_bulk_out_complete()`, `ubt_intr_complete()`, `ubt_isoc_in_complete()`, `ubt_isoc_out_complete()`, and `ubt_request_complete()`.

7.74.1.104 #define NG_SEND_MSG_HOOK(error, here, msg, hook, retaddr)

Value:

```
do {
    item_p _item;
    if ((_item = ng_package_msg(msg, NG_NOFLAGS)) == NULL) {
        (msg) = NULL;
        (error) = ENOMEM;
        break;
    }
    if (((error) = ng_address_hook((here), (_item),
        (hook), (retaddr))) == 0) {
        SAVE_LINE(_item);
        (error) = ng_snd_item((_item), 0);
    }
    (msg) = NULL;
} while (0)
```

Definition at line 892 of file `netgraph.h`.

Referenced by `cisco_input()`, `cisco_notify()`, `ng_atm_event_func()`, `ng_btsocket_l2cap_raw_control()`, `ng_btsocket_l2cap_raw_send_ngmsg()`, `ng_btsocket_l2cap_raw_send_sync_ngmsg()`, `ng_btsocket_l2cap_send_l2ca_cfg_req()`, `ng_btsocket_l2cap_send_l2ca_cfg_rsp()`, `ng_btsocket_l2cap_send_l2ca_con_req()`, `ng_btsocket_l2cap_send_l2ca_con_rsp_req()`, `ng_btsocket_l2cap_send_l2ca_discon_req()`, `ng_ether_link_state()`, `ng_hci_lp_con_cfm()`, `ng_hci_lp_con_ind()`, `ng_hci_lp_discon_ind()`, `ng_hci_lp_qos_cfm()`, `ng_hci_lp_qos_ind()`, `ng_hci_node_is_up()`, `ng_l2cap_l2ca_cfg_ind()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_ind()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_ind()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_qos_ind()`, `ng_l2cap_l2ca_write_rsp()`, `ng_l2cap_lp_con_ind()`, `ng_l2cap_lp_con_req()`, `ng_l2cap_lp_qos_req()`, `ng_l2cap_process_discon_timeout()`, `ng_l2cap_send_hook_info()`, `ng_one2many_notify()`, `ng_source_set_autosrc()`, `ng_vlan_rcvmsg()`, and `sync_con_queue()`.

7.74.1.105 #define NG_SEND_MSG_ID(error, here, msg, ID, retaddr)**Value:**

```

do {
    item_p _item;
    if ((_item = ng_package_msg(msg, NG_NOFLAGS)) == NULL) {\
        (msg) = NULL;
        (error) = ENOMEM;
        break;
    }
    if (((error) = ng_address_ID((here), (_item),
        (ID), (retaddr))) == 0) {
        SAVE_LINE(_item);
        (error) = ng_snd_item((_item), 0);
    }
    (msg) = NULL;
} while (0)

```

Definition at line 924 of file netgraph.h.

Referenced by ng_deflate_rcvdata(), ng_ksocket_finish_accept(), ng_ksocket_incoming2(), ng_l2tp_seq_failure(), ng_mppc_rcvdata(), ng_pppoe_connect(), ng_pred1_rcvdata(), ng_source_connect(), pppoe_send_event(), send_acname(), and send_sessionid().

7.74.1.106 #define NG_SEND_MSG_PATH(error, here, msg, path, retaddr)**Value:**

```

do {
    item_p _item;
    if ((_item = ng_package_msg(msg, NG_NOFLAGS)) == NULL) {\
        (msg) = NULL;
        (error) = ENOMEM;
        break;
    }
    if (((error) = ng_address_path((here), (_item),
        (path), (retaddr))) == 0) {
        SAVE_LINE(_item);
        (error) = ng_snd_item((_item), 0);
    }
    (msg) = NULL;
} while (0)

```

Definition at line 908 of file netgraph.h.

Referenced by ng_btsocket_hci_raw_control(), ng_btsocket_hci_raw_send_ngmsg(), and ng_btsocket_hci_raw_send_sync_ngmsg().

7.74.1.107 #define NG_SEPARATE_MALLOC

Definition at line 60 of file netgraph.h.

7.74.1.108 #define NG_TAG_PRIO 1

Definition at line 1118 of file netgraph.h.

Referenced by nglmi_inquire().

7.74.1.109 #define NG_WAITOK 0x00000002

Definition at line 1105 of file netgraph.h.

Referenced by ng_connect_data(), ng_getqblk(), ng_ksocket_incoming(), and ngd_send().

7.74.1.110 #define NG_WORKQ NGF_WORKQ

Definition at line 356 of file netgraph.h.

7.74.1.111 #define NGF_CLOSING 0x00000008

Definition at line 359 of file netgraph.h.

Referenced by ng_rmnode(), and ng_rmnode_self().

7.74.1.112 #define NGF_FORCE_WRITER 0x00000004

Definition at line 357 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.113 #define NGF_INVALID 0x00000001

Definition at line 353 of file netgraph.h.

Referenced by ng_rmnode(), and ng_rmnode_self().

7.74.1.114 #define NGF_REALLY_DIE 0x00000010

Definition at line 361 of file netgraph.h.

Referenced by ng_atm_shutdown(), ng_ether_shutdown(), ng_gif_shutdown(), and ng_xxx_shutdown().

7.74.1.115 #define NGF_TYPE1 0x10000000

Definition at line 363 of file netgraph.h.

7.74.1.116 #define NGF_TYPE2 0x20000000

Definition at line 364 of file netgraph.h.

7.74.1.117 #define NGF_TYPE3 0x40000000

Definition at line 365 of file netgraph.h.

7.74.1.118 #define NGF_TYPE4 0x80000000

Definition at line 366 of file netgraph.h.

7.74.1.119 #define NGF_WORKQ 0x00000002

Definition at line 355 of file netgraph.h.

Referenced by ng_setisr(), ng_worklist_remove(), and ngintr().

7.74.1.120 #define NGI_ARG1(i) _NGI_ARG1(i)

Definition at line 772 of file netgraph.h.

Referenced by ng_apply_item().

7.74.1.121 #define NGI_ARG2(i) _NGI_ARG2(i)

Definition at line 773 of file netgraph.h.

Referenced by ng_apply_item().

7.74.1.122 #define NGI_CLR_HOOK(i) _NGI_CLR_HOOK(i)

Definition at line 777 of file netgraph.h.

7.74.1.123 #define NGI_CLR_NODE(i) _NGI_CLR_NODE(i)

Definition at line 779 of file netgraph.h.

7.74.1.124 #define NGI_FN(i) _NGI_FN(i)

Definition at line 771 of file netgraph.h.

Referenced by ng_apply_item().

7.74.1.125 #define NGI_GET_HOOK(i, h)**Value:**

```
do {
    (h) = NGI_HOOK(i);
    _NGI_HOOK(i) = NULL;
} while (0)
```

Definition at line 804 of file netgraph.h.

Referenced by ng_apply_item(), ng_btsocket_l2cap_input(), and ng_btsocket_l2cap_raw_input().

7.74.1.126 #define NGI_GET_M(i, m)**Value:**

```
do {
    (m) = NGI_M(i);
    _NGI_M(i) = NULL;
} while (0)
```

Definition at line 786 of file netgraph.h.

Referenced by `ng_atm_rcvdata()`, `ng_atmpif_rcvdata()`, `ng_bridge_rcvdata()`, `ng_bt3c_rcvdata()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_hci_raw_node_rcvdata()`, `ng_btsocket_l2cap_input()`, `ng_ccatm_rcvdata()`, `ng_ccatm_rcvuni()`, `ng_deflate_rcvdata()`, `ng_device_rcvdata()`, `ng_eiface_rcvdata()`, `ng_etf_rcvdata()`, `ng_ether_rcvdata()`, `ng_gif_demux_rcvdata()`, `ng_gif_rcvdata()`, `ng_h4_rcvdata()`, `ng_hci_acl_rcvdata()`, `ng_hci_drv_rcvdata()`, `ng_hci_raw_rcvdata()`, `ng_hci_sco_rcvdata()`, `ng_iface_rcvdata()`, `ng_ipfw_rcvdata()`, `ng_ksocket_rcvdata()`, `ng_l2cap_rcvdata()`, `ng_l2tp_rcv_ctrl()`, `ng_l2tp_rcv_data()`, `ng_l2tp_rcv_lower()`, `ng_mppc_rcvdata()`, `ng_netflow_rcvdata()`, `ng_ppp_bypass()`, `ng_ppp_comp_rcv()`, `ng_ppp_comp_xmit()`, `ng_ppp_crypt_xmit()`, `ng_ppp_link_xmit()`, `ng_ppp_mp_rcv()`, `ng_ppp_mp_xmit()`, `ng_ppp_rcvdata()`, `ng_ppp_rcvdata_bypass()`, `ng_ppp_rcvdata_compress()`, `ng_ppp_rcvdata_decompress()`, `ng_ppp_rcvdata_decrypt()`, `ng_pppoe_rcvdata()`, `ng_pptpgre_rcv()`, `ng_pptpgre_xmit()`, `ng_pred1_rcvdata()`, `ng_rfc1490_rcvdata()`, `ng_source_rcvdata()`, `ng_sppp_rcvdata()`, `ng_sscfu_rcvlower()`, `ng_sscfu_rcvupper()`, `ng_sscop_rcvlower()`, `ng_sscop_rcvmanage()`, `ng_sscop_rcvupper()`, `ng_tcpmss_rcvdata()`, `ng_ubt_rcvdata()`, `ng_UI_rcvdata()`, `ng_unicast_rcvlower()`, `ng_unicast_rcvupper()`, `ng_vjc_rcvdata()`, `ng_vlan_rcvdata()`, `ng_xxx_rcvdata()`, `nga_rcv_async()`, `nga_rcv_sync()`, `ngfrm_decode()`, `ngfrm_rcvdata()`, `ngipi_rcvdata()`, `nglmi_rcvdata()`, `ngs_rcvdata()`, and `ngt_rcvdata()`.

7.74.1.127 #define NGI_GET_META(i, m)

Definition at line 1136 of file netgraph.h.

7.74.1.128 #define NGI_GET_MSG(i, m)

Value:

```
do {
    (m) = NGI_MSG(i);
    _NGI_MSG(i) = NULL;
} while (0)
```

Definition at line 792 of file netgraph.h.

Referenced by `cisco_rcvmsg()`, `ng_atm_rcvmsg()`, `ng_atmllc_rcvmsg()`, `ng_atmpif_rcvmsg()`, `ng_bpf_rcvmsg()`, `ng_bridge_rcvmsg()`, `ng_bt3c_rcvmsg()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_raw_input()`, `ng_ccatm_rcvmsg()`, `ng_deflate_rcvmsg()`, `ng_device_rcvmsg()`, `ng_eiface_rcvmsg()`, `ng_etf_rcvmsg()`, `ng_ether_rcvmsg()`, `ng_fec_rcvmsg()`, `ng_generic_msg()`, `ng_gif_demux_rcvmsg()`, `ng_gif_rcvmsg()`, `ng_h4_rcvmsg()`, `ng_hci_default_rcvmsg()`, `ng_hci_lp_acl_con_req()`, `ng_iface_rcvmsg()`, `ng_ksocket_rcvmsg()`, `ng_l2cap_default_rcvmsg()`, `ng_l2tp_rcvmsg()`, `ng_mppc_rcvmsg()`, `ng_nat_rcvmsg()`, `ng_netflow_rcvmsg()`, `ng_one2many_rcvmsg()`, `ng_ppp_rcvmsg()`, `ng_pppoe_rcvmsg()`, `ng_pptpgre_rcvmsg()`, `ng_pred1_rcvmsg()`, `ng_rfc1490_rcvmsg()`, `ng_source_rcvmsg()`, `ng_sppp_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_tag_rcvmsg()`, `ng_tcpmss_rcvmsg()`, `ng_ubt_rcvmsg()`, `ng_unicast_rcvmsg()`, `ng_vjc_rcvmsg()`, `ng_vlan_rcvmsg()`, `ng_xxx_rcvmsg()`, `nga_rcvmsg()`, `ng_rcvmsg()`, `ngh_rcvmsg()`, `nglmi_rcvmsg()`, `ngs_rcvmsg()`, and `ngt_rcvmsg()`.

7.74.1.129 #define NGI_GET_NODE(i, n)

Value:

```
/* YOU NOW HAVE THE REFERENCE */
do {
    (n) = NGI_NODE(i);
    _NGI_NODE(i) = NULL;
} while (0)
```

Definition at line 798 of file netgraph.h.

Referenced by ng_snd_item(), and ngintr().

7.74.1.130 #define NGI_HOOK(i) _NGI_HOOK(i)

Definition at line 775 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.131 #define NGI_M(i) _NGI_M(i)

Definition at line 768 of file netgraph.h.

Referenced by cisco_input(), cisco_rcvdata(), export_add(), export_send(), ng_atmllc_rcvdata(), ng_bpf_rcvdata(), ng_btsocket_hci_raw_node_rcvdata(), ng_gif_demux_rcvdata(), ng_hci_acl_rcvdata(), ng_hci_drv_rcvdata(), ng_hci_sco_rcvdata(), ng_hub_rcvdata(), ng_nat_rcvdata(), ng_one2many_rcvdata(), ng_package_data(), ng_ppp_bypass(), ng_ppp_comp_rcv(), ng_ppp_comp_xmit(), ng_ppp_crypt_rcv(), ng_ppp_crypt_xmit(), ng_ppp_mp_xmit(), ng_ppp_rcvdata(), ng_ppp_rcvdata_bypass(), ng_ppp_rcvdata_compress(), ng_ppp_rcvdata_decompress(), ng_ppp_rcvdata_decrypt(), ng_snd_item(), ng_tag_rcvdata(), ngh_rcvdata(), ngt_rcvdata(), and send_data_packets().

7.74.1.132 #define NGI_META(i) NULL

Definition at line 1134 of file netgraph.h.

7.74.1.133 #define NGI_MSG(i) _NGI_MSG(i)

Definition at line 769 of file netgraph.h.

Referenced by ng_apply_item(), ng_btsocket_hci_raw_node_rcvmsg(), ng_btsocket_l2cap_node_rcvmsg(), ng_btsocket_l2cap_raw_node_rcvmsg(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_req(), ng_hci_lp_con_rsp(), ng_hci_lp_discon_req(), ng_hci_lp_qos_req(), ng_hci_lp_sco_con_req(), ng_hci_upper_rcvmsg(), ng_l2cap_lower_rcvmsg(), ng_l2cap_upper_rcvmsg(), ng_package_msg(), ng_package_msg_self(), ng_ppp_rcvmsg(), ng_UI_rcvmsg(), and ngt_rcvmsg().

7.74.1.134 #define NGI_NODE(i) _NGI_NODE(i)

Definition at line 774 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.135 #define NGI_QUEUED_READER(i) ((i) → el_flags & NGQF_QREADER)

Definition at line 813 of file netgraph.h.

7.74.1.136 #define NGI_QUEUED_WRITER(i) (((i) → el_flags & NGQF_QMODE) == NGQF_QWRITER)

Definition at line 814 of file netgraph.h.

7.74.1.137 #define NGI_RETADDR(i) _NGI_RETADDR(i)

Definition at line 770 of file netgraph.h.

Referenced by `ng_deflate_rcvmsg()`, `ng_ksocket_rcvmsg()`, `ng_l2tp_rcvmsg()`, `ng_mppc_rcvmsg()`, `ng_package_msg()`, `ng_package_msg_self()`, `ng_pppoe_rcvmsg()`, `ng_pred1_rcvmsg()`, `ng_replace_retaddr()`, and `ngs_rcvmsg()`.

7.74.1.138 #define NGI_SET_HOOK(i, h) _NGI_SET_HOOK(i,h)

Definition at line 776 of file netgraph.h.

Referenced by `ng_address_hook()`, `ng_btsocket_l2cap_node_rcvdata()`, `ng_btsocket_l2cap_node_rcvmsg()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, and `ng_package_msg_self()`.

7.74.1.139 #define NGI_SET_NODE(i, n) _NGI_SET_NODE(i,n)

Definition at line 778 of file netgraph.h.

Referenced by `ng_address_hook()`, and `ng_package_msg_self()`.

7.74.1.140 #define NGI_SET_READER(i) ((i) → el_flags |= NGQF_QREADER)

Definition at line 811 of file netgraph.h.

Referenced by `ng_queue_rw()`.

7.74.1.141 #define NGI_SET_WRITER(i) ((i) → el_flags &= ~NGQF_QMODE)

Definition at line 810 of file netgraph.h.

Referenced by `ng_queue_rw()`.

7.74.1.142 #define NGQF_DATA 0x01

Definition at line 615 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_l2cap_input()`, `ng_free_item()`, `ng_package_data()`, and `ng_snd_item()`.

7.74.1.143 #define NGQF_FN 0x02

Definition at line 616 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_free_item()`, and `ng_snd_item()`.

7.74.1.144 #define NGQF_MESG 0x00

Definition at line 614 of file netgraph.h.

Referenced by `ng_apply_item()`, `ng_btsocket_hci_raw_input()`, `ng_btsocket_l2cap_input()`, `ng_btsocket_l2cap_raw_input()`, `ng_free_item()`, `ng_package_msg()`, `ng_package_msg_self()`, and `ng_snd_item()`.

7.74.1.145 #define NGQF_QMODE 0x08

Definition at line 623 of file netgraph.h.

7.74.1.146 #define NGQF_QREADER 0x08

Definition at line 624 of file netgraph.h.

7.74.1.147 #define NGQF_QWRITER 0x00

Definition at line 625 of file netgraph.h.

7.74.1.148 #define NGQF_READER 0x04

Definition at line 620 of file netgraph.h.

Referenced by ng_package_data(), ng_package_msg(), and ng_snd_item().

7.74.1.149 #define NGQF_RW 0x04

Definition at line 619 of file netgraph.h.

Referenced by ng_snd_item().

7.74.1.150 #define NGQF_TYPE 0x03

Definition at line 613 of file netgraph.h.

Referenced by ng_apply_item(), ng_btsocket_hci_raw_input(), ng_btsocket_l2cap_input(), ng_btsocket_l2cap_raw_input(), ng_free_item(), and ng_snd_item().

7.74.1.151 #define NGQF_UNDEF 0x03

Definition at line 617 of file netgraph.h.

Referenced by ng_free_item().

7.74.1.152 #define NGQF_WRITER 0x00

Definition at line 621 of file netgraph.h.

Referenced by ng_package_msg(), ng_package_msg_self(), and ng_snd_item().

7.74.1.153 #define SAVE_LINE(item) do {} while (0)

Definition at line 782 of file netgraph.h.

Referenced by ngc_send().

7.74.2 Typedef Documentation

7.74.2.1 typedef struct [ng_hook](#)* [hook_p](#)

Definition at line 86 of file netgraph.h.

7.74.2.2 typedef struct [ng_item](#)* [item_p](#)

Definition at line 84 of file netgraph.h.

7.74.2.3 typedef void* [meta_p](#)

Definition at line 1132 of file netgraph.h.

7.74.2.4 typedef void [ng_apply_t](#)(void *context, int error)

Definition at line 582 of file netgraph.h.

7.74.2.5 typedef int [ng_close_t](#)([node_p](#) node)

Definition at line 90 of file netgraph.h.

7.74.2.6 typedef int [ng_connect_t](#)([hook_p](#) hook)

Definition at line 94 of file netgraph.h.

7.74.2.7 typedef int [ng_constructor_t](#)([node_p](#) node)

Definition at line 89 of file netgraph.h.

7.74.2.8 typedef int [ng_disconnect_t](#)([hook_p](#) hook)

Definition at line 97 of file netgraph.h.

7.74.2.9 typedef [hook_p](#) [ng_findhook_t](#)([node_p](#) node, const char *name)

Definition at line 93 of file netgraph.h.

7.74.2.10 typedef int [ng_fn_eachhook](#)([hook_p](#) hook, void *arg)

Definition at line 395 of file netgraph.h.

7.74.2.11 typedef void [ng_item_fn](#)([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)

Definition at line 581 of file netgraph.h.

7.74.2.12 typedef int [ng_newhook_t](#)([node_p](#) node, [hook_p](#) hook, const char *name)

Definition at line 92 of file netgraph.h.

7.74.2.13 typedef int [ng_rcvdata_t](#)([hook_p](#) hook, [item_p](#) item)

Definition at line 96 of file netgraph.h.

7.74.2.14 typedef int [ng_rcvitem](#)([node_p](#) node, [hook_p](#) hook, [item_p](#) item)

Definition at line 98 of file netgraph.h.

7.74.2.15 typedef int [ng_rcvmsg_t](#)([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)

Definition at line 95 of file netgraph.h.

7.74.2.16 typedef int [ng_shutdown_t](#)([node_p](#) node)

Definition at line 91 of file netgraph.h.

7.74.2.17 typedef struct [ng_node](#)* [node_p](#)

Definition at line 85 of file netgraph.h.

7.74.3 Function Documentation

7.74.3.1 MALLOC_DECLARE (M_NETGRAPH_MSG)

7.74.3.2 MALLOC_DECLARE (M_NETGRAPH)

7.74.3.3 int [ng_address_hook](#) ([node_p](#) here, [item_p](#) item, [hook_p](#) hook, [ng_ID_t](#) retaddr)

Definition at line 3437 of file ng_base.c.

References [ITEM_DEBUG_CHECKS](#), [NG_FREE_ITEM](#), [NG_HOOK_NOT_VALID](#), [NG_HOOK_PEER](#), [NG_HOOK_REF](#), [NG_NODE_NOT_VALID](#), [NG_NODE_REF](#), [NG_PEER_NODE](#), [NGI_SET_HOOK](#), [NGI_SET_NODE](#), [SET_RETADDR](#), and [TRAP_ERROR](#).

7.74.3.4 int [ng_address_ID](#) ([node_p](#) here, [item_p](#) item, [ng_ID_t](#) ID, [ng_ID_t](#) retaddr)

Definition at line 3498 of file ng_base.c.

7.74.3.5 int [ng_address_path](#) ([node_p](#) here, [item_p](#) item, char * address, [ng_ID_t](#) raddr)

Definition at line 3472 of file ng_base.c.

Referenced by [ng_connect_data\(\)](#), and [ngc_send\(\)](#).

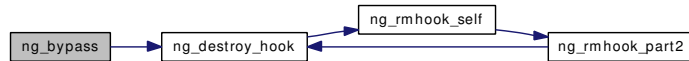
7.74.3.6 `int ng_bypass (hook_p hook1, hook_p hook2)`

Definition at line 1120 of file `ng_base.c`.

References `ng_hook::hk_node`, `ng_hook::hk_peer`, `ng_deadhook`, `ng_destroy_hook()`, `NG_HOOK_UNREF`, and `TRAP_ERROR`.

Referenced by `ngt_close()`.

Here is the call graph for this function:



7.74.3.7 `int ng_callout (struct callout * c, node_p node, hook_p hook, int ticks, ng_item_fn * fn, void * arg1, int arg2)`

Definition at line 3588 of file `ng_base.c`.

Referenced by `cisco_keepalive()`, `cisco_newhook()`, `LMI_ticker()`, `ng_bridge_constructor()`, `ng_h4_timeout()`, `ng_hci_command_timeout()`, `ng_hci_con_timeout()`, `ng_l2cap_command_timeout()`, `ng_l2cap_discon_timeout()`, `ng_l2cap_lp_timeout()`, `ng_l2tp_recv_ctrl()`, `ng_l2tp_seq_rack_timeout()`, `ng_l2tp_seq_recv_ns()`, `ng_ppp_start_frag_timer()`, `ng_pppoe_sendpacket()`, `ng_pptpgre_start_recv_ack_timer()`, `ng_pptpgre_start_send_ack_timer()`, `ng_source_intr()`, `ng_source_start()`, `nglmi_startup()`, `ngt_start()`, and `pppoe_ticker()`.

7.74.3.8 `hook_p ng_findhook (node_p node, const char * name)`

Definition at line 1008 of file `ng_base.c`.

References `ng_type::findhook`, `ng_node::nd_type`, `NG_HOOK_IS_VALID`, and `NG_HOOK_NAME`.

Referenced by `ng_add_hook()`, `ng_bpf_rcvdata()`, `ng_bpf_rcvmsg()`, `ng_con_nodes()`, `ng_con_part2()`, `ng_etf_rcvmsg()`, `ng_generic_msg()`, `ng_path2noderef()`, `ng_tag_rcvmsg()`, `ng_tag_setdata_in()`, `ng_tcpmss_rcvmsg()`, `ng_vlan_rcvmsg()`, `ngd_send()`, and `ng_h_rcvmsg()`.

7.74.3.9 `struct ng_type* ng_findtype (const char * type)`

Definition at line 1199 of file `ng_base.c`.

References `ng_type::name`.

Referenced by `ng_make_node()`, `ng_make_node_common()`, `ng_newtype()`, and `ngc_send()`.

7.74.3.10 `void ng_free_item (item_p item)`

Definition at line 2947 of file `ng_base.c`.

References `_NGI_ARG1`, `_NGI_ARG2`, `_NGI_CLR_HOOK`, `_NGI_CLR_NODE`, `_NGI_FN`, `_NGI_M`, `_NGI_MSG`, `_NGI_RETADDR`, `ng_item::apply`, `ng_item::el_flags`, `NG_FREE_M`, `NG_FREE_MSG`, `NGQF_DATA`, `NGQF_FN`, `NGQF_MESG`, `NGQF_TYPE`, and `NGQF_UNDEF`.

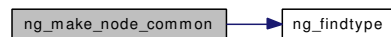
7.74.3.11 `int ng_make_node_common (struct ng_type * typep, node_p * nodep)`

Definition at line 588 of file `ng_base.c`.

References `ng_queue::last`, `ng_type::name`, `ng_node::nd_ID`, `ng_node::nd_input_queue`, `ng_node::nd_type`, `NG_ALLOC_NODE`, `ng_findtype()`, `NG_IDHASH_FIND`, `NG_IDHASH_FN`, `NG_NODE_REF`, `ng_queue::q_flags`, `ng_queue::q_mtx`, `ng_queue::q_node`, `ng_queue::queue`, `ng_type::refs`, and `TRAP_ERROR`.

Referenced by `bt3c_pccard_attach()`, `ng_atm_attach()`, `ng_attach_cntl()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_ether_attach()`, `ng_gif_attach()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_ipfw_mod_event()`, `ng_ksocket_finish_accept()`, `ng_make_node()`, `ng_ubt_shutdown()`, and `ngt_open()`.

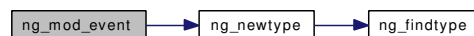
Here is the call graph for this function:

**7.74.3.12** `int ng_mod_event (module_t mod, int what, void * arg)`

Definition at line 2996 of file `ng_base.c`.

References `ng_type::mod_event`, `ng_newtype()`, and `ng_type::refs`.

Here is the call graph for this function:

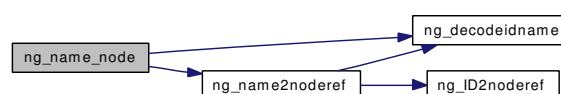
**7.74.3.13** `int ng_name_node (node_p node, const char * name)`

Definition at line 799 of file `ng_base.c`.

References `ng_decodeidname()`, `ng_name2noderef()`, `NG_NODE_NAME`, `NG_NODE_UNREF`, `NG_NODESIZ`, and `TRAP_ERROR`.

Referenced by `bt3c_pccard_attach()`, `ng_atm_attach()`, `ng_bind()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_eiface_constructor()`, `ng_ether_attach()`, `ng_fec_constructor()`, `ng_generic_msg()`, `ng_gif_attach()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_iface_constructor()`, `ng_ipfw_mod_event()`, `ng_sppp_constructor()`, `ng_ubt_shutdown()`, and `ngt_open()`.

Here is the call graph for this function:



7.74.3.14 `int ng_newtype (struct ng_type * tp)`

Definition at line 1145 of file ng_base.c.

References `ng_type::name`, `NG_ABI_VERSION`, `ng_findtype()`, `NG_TYPESIZ`, `ng_type::refs`, `TRAP_ERROR`, and `ng_type::version`.

Referenced by `bt3c_modevent()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_raw_init()`, `ng_mod_event()`, and `ubt_modevent()`.

Here is the call graph for this function:

**7.74.3.15** `ng_ID_t ng_node2ID (node_p node)`

Definition at line 786 of file ng_base.c.

References `NG_NODE_ID`.

Referenced by `ng_bridge_nodename()`, `ng_generic_msg()`, `ng_h4_ioctl()`, `ng_package_msg_self()`, and `ngt_ioctl()`.

7.74.3.16 `item_p ng_package_data (struct mbuf * m, int flags)`

Definition at line 3373 of file ng_base.c.

References `ng_item::el_flags`, `ng_item::el_next`, `ITEM_DEBUG_CHECKS`, `NG_FREE_M`, `ng_getqblk()`, `NGI_M`, `NGQF_DATA`, and `NGQF_READER`.

Referenced by `get_export_dgram()`, `ng_connect_data()`, `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, and `ng_ppp_mp_xmit()`.

Here is the call graph for this function:

**7.74.3.17** `item_p ng_package_msg (struct ng_mesg * msg, int flags)`

Definition at line 3396 of file ng_base.c.

References `ng_mesg::ng_msghdr::cmd`, `ng_item::el_flags`, `ng_item::el_next`, `ng_mesg::header`, `ITEM_DEBUG_CHECKS`, `NG_FREE_MSG`, `ng_getqblk()`, `NGI_MSG`, `NGI_RETADDR`, `NGM_READONLY`, `NGQF_MESG`, `NGQF_READER`, and `NGQF_WRITER`.

Referenced by `ngc_send()`.

Here is the call graph for this function:

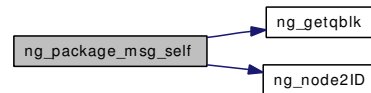


7.74.3.18 `item_p ng_package_msg_self (node_p here, hook_p hook, struct ng_mesg * msg)`

Definition at line 3525 of file ng_base.c.

References `ng_item::el_flags`, `ng_item::el_next`, `NG_FREE_MSG`, `ng_getqblk()`, `NG_HOOK_REF`, `ng_node2ID()`, `NG_NODE_REF`, `NG_NOFLAGS`, `NGI_MSG`, `NGI_RETADDR`, `NGI_SET_HOOK`, `NGI_SET_NODE`, `NGQF_MESG`, and `NGQF_WRITER`.

Here is the call graph for this function:

**7.74.3.19** `void ng_replace_retaddr (node_p here, item_p item, ng_ID_t retaddr)`

Definition at line 3644 of file ng_base.c.

References `NGI_RETADDR`.

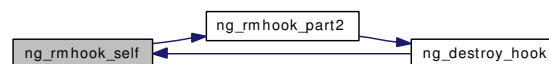
7.74.3.20 `int ng_rmhook_self (hook_p hook)`

Definition at line 1492 of file ng_base.c.

References `ng_deadnode`, `NG_HOOK_NODE`, `ng_rmhook_part2()`, and `ng_send_fn`.

Referenced by `ng_destroy_hook()`, `ng_pppoe_rcvdata()`, and `pppoe_ticker()`.

Here is the call graph for this function:

**7.74.3.21** `int ng_rmnode_self (node_p here)`

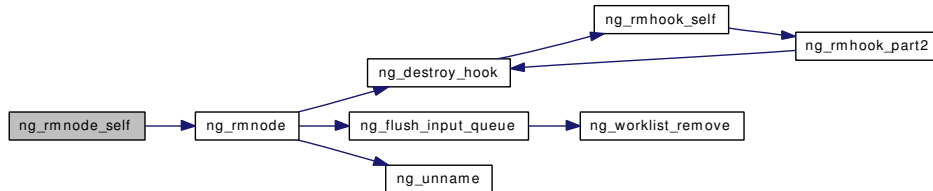
Definition at line 1470 of file ng_base.c.

References `ng_node::nd_flags`, `ng_deadnode`, `ng_rmnode()`, `ng_send_fn`, `NGF_CLOSING`, and `NGF_INVALID`.

Referenced by `bt3c_pccard_detach()`, `cisco_disconnect()`, `ng_atm_detach()`, `ng_atmllc_disconnect()`, `ng_bpf_disconnect()`, `ng_bridge_disconnect()`, `ng_ccatm_disconnect()`, `ng_deflate_disconnect()`, `ng_device_disconnect()`, `ng_ether_detach()`, `ng_ether_disconnect()`, `ng_gif_detach()`, `ng_gif_disconnect()`, `ng_h4_close()`, `ng_hci_disconnect()`, `ng_hub_disconnect()`, `ng_ksocket_disconnect()`, `ng_ksocket_shutdown()`, `ng_l2cap_disconnect()`, `ng_l2tp_disconnect()`, `ng_mppc_disconnect()`, `ng_nat_disconnect()`, `ng_netflow_disconnect()`, `ng_one2many_disconnect()`, `ng_ppp_disconnect()`, `ng_pppoe_disconnect()`, `ng_pptpgre_disconnect()`, `ng_pred1_disconnect()`, `ng_rfc1490_disconnect()`, `ng_socket_free_priv()`, `ng_source_disconnect()`, `ng_split_disconnect()`, `ng_sscfu_disconnect()`, `ng_sscop_disconnect()`, `ng_tag_disconnect()`, `ng_tcpmss_disconnect()`, `ng_UI_disconnect()`, `ng_uni_disconnect()`, `ng_vjc_disconnect()`, `ng_vlan_disconnect()`, `ng_xxx_disconnect()`, `nga_disconnect()`, `nge_disconnect()`, `ngfrm_disconnect()`, `ngh_`

disconnect(), ngipi_disconnect(), nglmi_disconnect(), ngs_disconnect(), ngt_close(), ngt_disconnect(), and USB_DETACH().

Here is the call graph for this function:



7.74.3.22 int ng_rmtree (struct ng_type * tp)

Definition at line 1180 of file ng_base.c.

References ng_type::refs, and TRAP_ERROR.

Referenced by bt3c_modevent(), and ubt_modevent().

7.74.3.23 int ng_send_fn1 (node_p node, hook_p hook, ng_item_fn * fn, void * arg1, int arg2, int flags)

Definition at line 3554 of file ng_base.c.

Referenced by ng_ksocket_incoming().

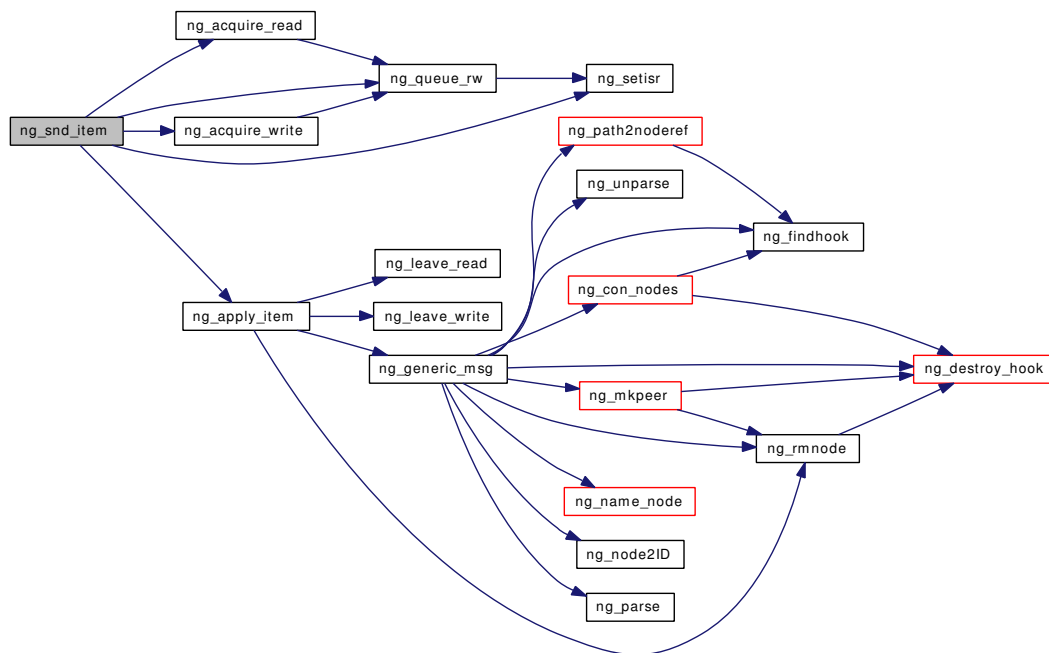
7.74.3.24 int ng_snd_item (item_p item, int queue)

Definition at line 2170 of file ng_base.c.

References CHECK_DATA_MBUF, ng_item::el_flags, ng_hook::hk_flags, HK_FORCE_WRITER, HK_QUEUE, ng_node::nd_flags, ng_node::nd_input_queue, NEXT_QUEUED_ITEM_CAN_PROCEED, ng_acquire_read(), ng_acquire_write(), ng_apply_item(), NG_FREE_ITEM, NG_HOOK_NODE, NG_HOOK_NOT_VALID, NG_NODE_NOT_VALID, NG_NODE_UNREF, NG_PROGRESS, NG_QUEUE, ng_queue_rw(), ng_setisr(), NGF_FORCE_WRITER, NGI_GET_NODE, NGI_HOOK, NGI_M, NGI_NODE, NGQF_DATA, NGQF_FN, NGQF_MESG, NGQF_READER, NGQF_RW, NGQF_TYPE, NGQF_WRITER, NGQRW_R, NGQRW_W, ng_queue::q_mtx, ng_queue::q_node, and TRAP_ERROR.

Referenced by ng_callout_trampoline(), and ngc_send().

Here is the call graph for this function:



7.74.3.25 `int ng_uncallout (struct callout * c, node_p node)`

Definition at line 3615 of file `ng_base.c`.

Referenced by `cisco_disconnect()`, `ng_bridge_shutdown()`, `ng_h4_untimeout()`, `ng_hci_command_untimeout()`, `ng_hci_con_untimeout()`, `ng_l2cap_command_untimeout()`, `ng_l2cap_discon_untimeout()`, `ng_l2cap_lp_untimeout()`, `ng_l2tp_seq_reset()`, `ng_l2tp_shutdown()`, `ng_l2tp_xmit_ctrl()`, `ng_ppp_stop_frag_timer()`, `ng_pppoe_disconnect()`, `ng_pppoe_rcvdata()`, `ng_pptpgre_stop_recv_ack_timer()`, `ng_pptpgre_stop_send_ack_timer()`, `ng_source_stop()`, `nglmi_disconnect()`, and `ngt_close()`.

7.74.3.26 `void ng_unref_hook (hook_p hook)`

Definition at line 922 of file `ng_base.c`.

References `_NG_HOOK_NODE`, `_NG_NODE_UNREF`, `ng_hook::hk_refs`, `ng_deadhook`, and `NG_FREE_HOOK`.

7.74.3.27 `int ng_unref_node (node_p node)`

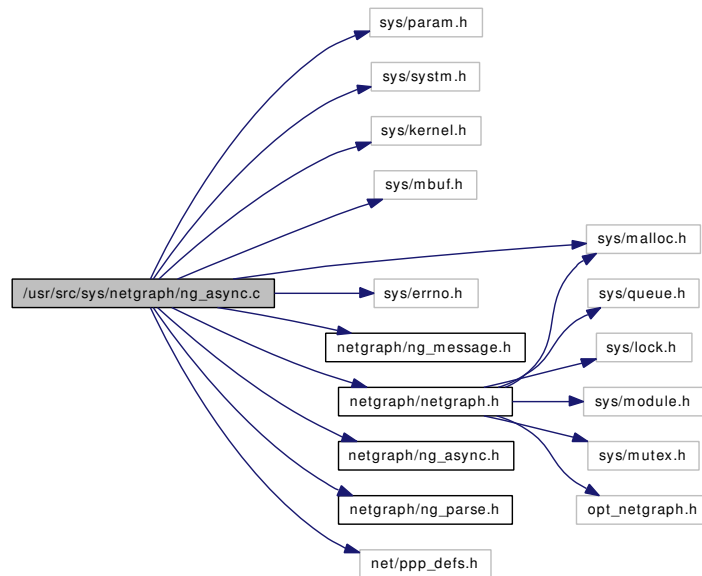
Definition at line 741 of file `ng_base.c`.

References `ng_node::nd_input_queue`, `ng_node::nd_refs`, `ng_node::nd_type`, `ng_deadnode`, `NG_FREE_NODE`, `ng_queue::q_mtx`, and `ng_type::refs`.

7.75 /usr/src/sys/netgraph/ng_async.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_async.h>
#include <netgraph/ng_parse.h>
#include <net/ppp_defs.h>
```

Include dependency graph for ng_async.c:



Data Structures

- struct [ng_async_private](#)

Defines

- #define [M_NETGRAPH_ASYNC](#) M_NETGRAPH
- #define [MODE_HUNT](#) 0
- #define [MODE_NORMAL](#) 1
- #define [MODE_ESC](#) 2
- #define [ASYNC_BUF_SIZE](#)(smru) (2 * (smru) + 10)

- #define [SYNC_BUF_SIZE](#)(amru) ((amru) + 10)
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)
- #define [ADD_BYTE](#)(x) nga_async_add(sc, &fcs, accm, &alen, (x))

Typedefs

- typedef [ng_async_private](#) * [sc_p](#)

Functions

- static int [nga_rcv_sync](#) (const [sc_p](#) sc, [item_p](#) item)
- static int [nga_rcv_async](#) (const [sc_p](#) sc, [item_p](#) item)
- [NETGRAPH_INIT](#) (async,&typestruct)
- static int [nga_constructor](#) ([node_p](#) node)
- static int [nga_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *[name](#))
- static int [nga_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [nga_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [nga_shutdown](#) ([node_p](#) node)
- static int [nga_disconnect](#) ([hook_p](#) hook)
- static __inline void [nga_async_add](#) (const [sc_p](#) sc, u_int16_t *fcs, u_int32_t accm, int *len, u_char x)

Variables

- static [ng_constructor_t](#) [nga_constructor](#)
- static [ng_rcvdata_t](#) [nga_rcvdata](#)
- static [ng_rcvmsg_t](#) [nga_rcvmsg](#)
- static [ng_shutdown_t](#) [nga_shutdown](#)
- static [ng_newhook_t](#) [nga_newhook](#)
- static [ng_disconnect_t](#) [nga_disconnect](#)
- static struct [ng_parse_struct_field](#) [nga_config_type_fields](#) [] = [NG_ASYNC_CONFIG_TYPE_INFO](#)
- static struct [ng_parse_type](#) [nga_config_type](#)
- static struct [ng_parse_struct_field](#) [nga_stats_type_fields](#) [] = [NG_ASYNC_STATS_TYPE_INFO](#)
- static struct [ng_parse_type](#) [nga_stats_type](#)
- static struct [ng_cmdlist](#) [nga_cmdlist](#) []
- static struct [ng_type](#) typestruct
- static const u_int16_t [fcstab](#) []

7.75.1 Define Documentation

7.75.1.1 #define [ADD_BYTE](#)(x) [nga_async_add](#)(sc, &fcs, accm, &alen, (x))

Referenced by [nga_rcv_sync](#)().

7.75.1.2 #define [ASYNC_BUF_SIZE](#)(smru) (2 * (smru) + 10)

Definition at line 92 of file [ng_async.c](#).

Referenced by [nga_constructor](#)(), and [nga_rcvmsg](#)().

7.75.1.3 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 94 of file ng_async.c.

7.75.1.4 #define M_NETGRAPH_ASYNC M_NETGRAPH

Definition at line 66 of file ng_async.c.

Referenced by nga_constructor(), nga_rcvmsg(), and nga_shutdown().

7.75.1.5 #define MODE_ESC 2

Definition at line 73 of file ng_async.c.

7.75.1.6 #define MODE_HUNT 0

Definition at line 71 of file ng_async.c.

Referenced by nga_constructor(), and nga_rcvmsg().

7.75.1.7 #define MODE_NORMAL 1

Definition at line 72 of file ng_async.c.

7.75.1.8 #define SYNC_BUF_SIZE(amru) ((amru) + 10)

Definition at line 93 of file ng_async.c.

Referenced by nga_constructor(), and nga_rcvmsg().

7.75.2 Typedef Documentation**7.75.2.1 typedef struct [ng_async_private](#)* [sc_p](#)**

Definition at line 89 of file ng_async.c.

7.75.3 Function Documentation**7.75.3.1 NETGRAPH_INIT (async, & *typestruct*)****7.75.3.2 static `__inline` void [nga_async_add](#) (const [sc_p](#) *sc*, `u_int16_t` * *fcs*, `u_int32_t` *accm*, `int` * *len*, `u_char` *x*) [*static*]**

Definition at line 395 of file ng_async.c.

References [ng_async_private::abuf](#).

7.75.3.3 static int nga_constructor (node_p node) [static]

Definition at line 182 of file ng_async.c.

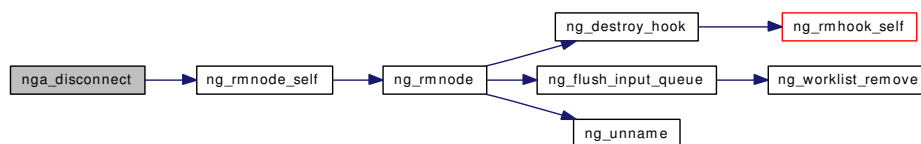
References ASYNC_BUF_SIZE, M_NETGRAPH_ASYNC, MODE_HUNT, NG_ASYNC_DEFAULT_MRU, NG_NODE_SET_PRIVATE, and SYNC_BUF_SIZE.

7.75.3.4 static int nga_disconnect (hook_p hook) [static]

Definition at line 365 of file ng_async.c.

References ng_async_private::async, ng_async_private::lasttime, NG_HOOK_NODE, NG_NODE_IS_INVALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), ng_async_private::stats, and ng_async_private::sync.

Here is the call graph for this function:

**7.75.3.5 static int nga_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 214 of file ng_async.c.

References ng_async_private::async, NG_ASYNC_HOOK_ASYNC, NG_ASYNC_HOOK_SYNC, NG_HOOK_FORCE_WRITER, NG_NODE_PRIVATE, and ng_async_private::sync.

7.75.3.6 static int nga_rcv_async (const sc_p sc, item_p item) [static]

Definition at line 503 of file ng_async.c.

References ng_async_cfg::amru, ng_async_stat::asyncBadCheckSums, ng_async_stat::asyncOctets, ng_async_stat::asyncOverflows, ng_async_stat::asyncRunts, ng_async_private::cfg, ng_async_cfg::enabled, ng_async_private::fcs, NG_FWD_ITEM_HOOK, NG_FWD_NEW_DATA, NG_SEND_DATA_ONLY, NGL_GET_M, PPP_GOODFCS, ng_async_private::sbuf, ng_async_private::slen, ng_async_private::stats, and ng_async_private::sync.

Referenced by nga_rcvdata().

7.75.3.7 static int nga_rcv_sync (const sc_p sc, item_p item) [static]

Definition at line 411 of file ng_async.c.

References ng_async_private::abuf, ng_async_cfg::accm, ADD_BYTE, ng_async_private::async, ng_async_private::cfg, ng_async_cfg::enabled, ng_async_private::lasttime, NG_FREE_ITEM, NG_FREE_M, NG_FWD_ITEM_HOOK, NG_FWD_NEW_DATA, NGL_GET_M, PPP_INITFCS, ng_async_cfg::smru, ng_async_private::stats, ng_async_stat::syncFrames, ng_async_stat::syncOctets, and ng_async_stat::syncOverflows.

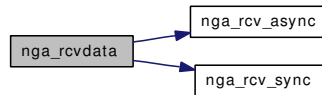
Referenced by nga_rcvdata().

7.75.3.8 `static int nga_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 251 of file `ng_async.c`.

References `ng_async_private::async`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `nga_rcv_async()`, `nga_rcv_sync()`, and `ng_async_private::sync`.

Here is the call graph for this function:



7.75.3.9 `static int nga_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 266 of file `ng_async.c`.

References `ng_async_private::abuf`, `ng_async_private::amode`, `ng_async_cfg::amru`, `ng_mesg::ng_msghdr::arglen`, `ASYNC_BUF_SIZE`, `ng_async_private::cfg`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ERROUT`, `ng_mesg::header`, `M_NETGRAPH_ASYNC`, `MODE_HUNT`, `NG_ASYNC_MAX_MRU`, `NG_ASYNC_MIN_MRU`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_ASYNC_CMD_CLR_STATS`, `NGM_ASYNC_CMD_GET_CONFIG`, `NGM_ASYNC_CMD_GET_STATS`, `NGM_ASYNC_CMD_SET_CONFIG`, `NGM_ASYNC_COOKIE`, `ng_async_private::sbuf`, `ng_async_private::slen`, `ng_async_cfg::smru`, `ng_async_private::stats`, `SYNC_BUF_SIZE`, and `ng_mesg::ng_msghdr::typecookie`.

7.75.3.10 `static int nga_shutdown (node_p node)` [static]

Definition at line 348 of file `ng_async.c`.

References `ng_async_private::abuf`, `M_NETGRAPH_ASYNC`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `ng_async_private::sbuf`.

7.75.4 Variable Documentation

7.75.4.1 `static const u_int16_t fcstab` [static]

Definition at line 172 of file `ng_async.c`.

7.75.4.2 `struct ng_cmdlist nga_cmdlist[]` [static]

Definition at line 125 of file `ng_async.c`.

7.75.4.3 `struct ng_parse_type nga_config_type` [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &nga_config_type_fields
}
  
```

Definition at line 111 of file ng_async.c.

7.75.4.4 `struct ng_parse_struct_field nga_config_type_fields[] = NG_ASYNC_CONFIG_TYPE_ -
INFO [static]`

Definition at line 110 of file ng_async.c.

7.75.4.5 `ng_constructor_t nga_constructor [static]`

Definition at line 97 of file ng_async.c.

7.75.4.6 `ng_disconnect_t nga_disconnect [static]`

Definition at line 102 of file ng_async.c.

7.75.4.7 `ng_newhook_t nga_newhook [static]`

Definition at line 101 of file ng_async.c.

7.75.4.8 `ng_rcvdata_t nga_rcvdata [static]`

Definition at line 98 of file ng_async.c.

7.75.4.9 `ng_rcvmsg_t nga_rcvmsg [static]`

Definition at line 99 of file ng_async.c.

7.75.4.10 `ng_shutdown_t nga_shutdown [static]`

Definition at line 100 of file ng_async.c.

7.75.4.11 `struct ng_parse_type nga_stats_type [static]`

Initial value:

```
{  
    &ng_parse_struct_type,  
    &nga_stats_type_fields  
}
```

Definition at line 119 of file ng_async.c.

7.75.4.12 `struct ng_parse_struct_field nga_stats_type_fields[] = NG_ASYNC_STATS_TYPE_ -
INFO [static]`

Definition at line 118 of file ng_async.c.

7.75.4.13 struct `ng_type_t` typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_ASYNC_NODE_TYPE,
    .constructor =  nga_constructor,
    .rcvmsg =      nga_rcvmsg,
    .shutdown =    nga_shutdown,
    .newhook =     nga_newhook,
    .rcvdata =     nga_rcvdata,
    .disconnect =  nga_disconnect,
    .cmdlist =     nga_cmdlist
}
```

Definition at line 158 of file `ng_async.c`.

7.76 /usr/src/sys/netgraph/ng_async.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_async_stat](#)
- struct [ng_async_cfg](#)

Defines

- #define [NG_ASYNC_NODE_TYPE](#) "async"
- #define [NGM_ASYNC_COOKIE](#) 886473717
- #define [NG_ASYNC_HOOK_SYNC](#) "sync"
- #define [NG_ASYNC_HOOK_ASYNC](#) "async"
- #define [NG_ASYNC_MIN_MRU](#) 1
- #define [NG_ASYNC_MAX_MRU](#) 8192
- #define [NG_ASYNC_DEFAULT_MRU](#) 1600
- #define [NG_ASYNC_STATS_TYPE_INFO](#)
- #define [NG_ASYNC_CONFIG_TYPE_INFO](#)

Enumerations

- enum { [NGM_ASYNC_CMD_GET_STATS](#) = 1, [NGM_ASYNC_CMD_CLR_STATS](#), [NGM_ASYNC_CMD_SET_CONFIG](#), [NGM_ASYNC_CMD_GET_CONFIG](#) }

7.76.1 Define Documentation

7.76.1.1 #define NG_ASYNC_CONFIG_TYPE_INFO

Value:

```

{
    { "enabled",          &ng_parse_int8_type    }, \
    { "amru",            &ng_parse_uint16_type   }, \
    { "smru",            &ng_parse_uint16_type   }, \
    { "accm",            &ng_parse_hint32_type   }, \
    { NULL }
}
  
```

Definition at line 94 of file [ng_async.h](#).

7.76.1.2 #define NG_ASYNC_DEFAULT_MRU 1600

Definition at line 58 of file [ng_async.h](#).

Referenced by [nga_constructor\(\)](#).

7.76.1.3 #define NG_ASYNC_HOOK_ASYNC "async"

Definition at line 53 of file ng_async.h.

Referenced by nga_newhook().

7.76.1.4 #define NG_ASYNC_HOOK_SYNC "sync"

Definition at line 52 of file ng_async.h.

Referenced by nga_newhook().

7.76.1.5 #define NG_ASYNC_MAX_MRU 8192

Definition at line 57 of file ng_async.h.

Referenced by nga_rcvmsg().

7.76.1.6 #define NG_ASYNC_MIN_MRU 1

Definition at line 56 of file ng_async.h.

Referenced by nga_rcvmsg().

7.76.1.7 #define NG_ASYNC_NODE_TYPE "async"

Definition at line 48 of file ng_async.h.

7.76.1.8 #define NG_ASYNC_STATS_TYPE_INFO**Value:**

```
{
    \
    { "syncOctets",      &ng_parse_uint32_type }, \
    { "syncFrames",    &ng_parse_uint32_type }, \
    { "syncOverflows", &ng_parse_uint32_type }, \
    { "asyncOctets",   &ng_parse_uint32_type }, \
    { "asyncFrames",   &ng_parse_uint32_type }, \
    { "asyncRunts",    &ng_parse_uint32_type }, \
    { "asyncOverflows", &ng_parse_uint32_type }, \
    { "asyncBadChecksums", &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 73 of file ng_async.h.

7.76.1.9 #define NGM_ASYNC_COOKIE 886473717

Definition at line 49 of file ng_async.h.

Referenced by nga_rcvmsg().

7.76.2 Enumeration Type Documentation

7.76.2.1 anonymous enum

Enumerator:

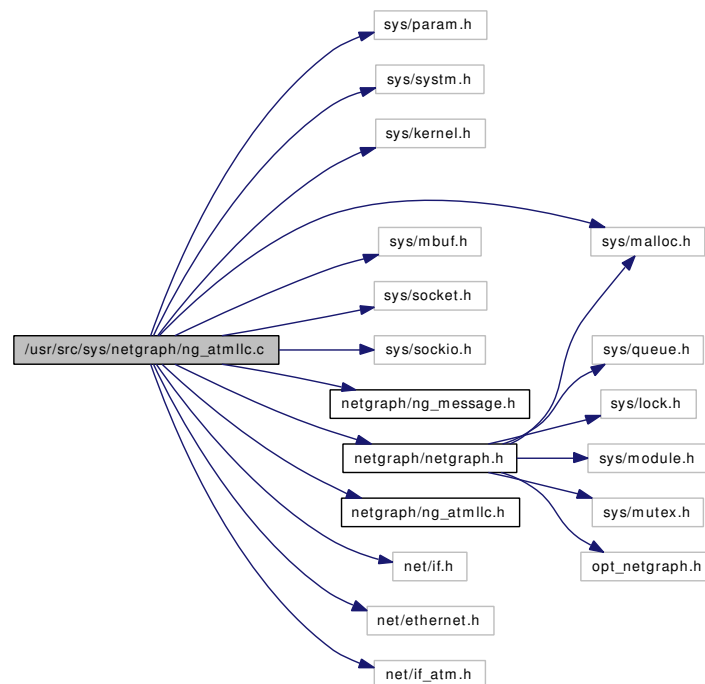
NGM_ASYNC_CMD_GET_STATS
NGM_ASYNC_CMD_CLR_STATS
NGM_ASYNC_CMD_SET_CONFIG
NGM_ASYNC_CMD_GET_CONFIG

Definition at line 103 of file ng_async.h.

7.77 /usr/src/sys/netgraph/ng_atmllc.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/sockio.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_atmllc.h>
#include <net/if.h>
#include <net/ethernet.h>
#include <net/if_atm.h>
```

Include dependency graph for ng_atmllc.c:



Data Structures

- struct [ng_atmllc_priv](#)

Defines

- #define `NG_ATMLLC_HEADER` "\252\252\3\0\200\302"
- #define `NG_ATMLLC_HEADER_LEN` (sizeof(struct atmllc))
- #define `NG_ATMLLC_TYPE_ETHERNET_FCS` 0x0001
- #define `NG_ATMLLC_TYPE_FDDI_FCS` 0x0004
- #define `NG_ATMLLC_TYPE_ETHERNET_NOFCS` 0x0007
- #define `NG_ATMLLC_TYPE_FDDI_NOFCS` 0x000A

Functions

- `NETGRAPH_INIT` (atmllc,&ng_atmllc_typestruct)
- static int `ng_atmllc_constructor` (node_p node)
- static int `ng_atmllc_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_atmllc_shutdown` (node_p node)
- static int `ng_atmllc_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_atmllc_rcvdata` (hook_p hook, item_p item)
- static int `ng_atmllc_disconnect` (hook_p hook)

Variables

- static `ng_constructor_t` ng_atmllc_constructor
- static `ng_shutdown_t` ng_atmllc_shutdown
- static `ng_rcvmsg_t` ng_atmllc_rcvmsg
- static `ng_newhook_t` ng_atmllc_newhook
- static `ng_rcvdata_t` ng_atmllc_rcvdata
- static `ng_disconnect_t` ng_atmllc_disconnect
- static struct `ng_type` ng_atmllc_typestruct

7.77.1 Define Documentation

7.77.1.1 #define `NG_ATMLLC_HEADER` "\252\252\3\0\200\302"

Definition at line 45 of file ng_atmllc.c.

Referenced by ng_atmllc_rcvdata().

7.77.1.2 #define `NG_ATMLLC_HEADER_LEN` (sizeof(struct atmllc))

Definition at line 46 of file ng_atmllc.c.

Referenced by ng_atmllc_rcvdata().

7.77.1.3 #define `NG_ATMLLC_TYPE_ETHERNET_FCS` 0x0001

Definition at line 47 of file ng_atmllc.c.

Referenced by ng_atmllc_rcvdata().

7.77.1.4 #define NG_ATMLLC_TYPE_ETHERNET_NOFCS 0x0007

Definition at line 49 of file ng_atmlc.c.

Referenced by ng_atmlc_rcvdata().

7.77.1.5 #define NG_ATMLLC_TYPE_FDDI_FCS 0x0004

Definition at line 48 of file ng_atmlc.c.

Referenced by ng_atmlc_rcvdata().

7.77.1.6 #define NG_ATMLLC_TYPE_FDDI_NOFCS 0x000A

Definition at line 50 of file ng_atmlc.c.

Referenced by ng_atmlc_rcvdata().

7.77.2 Function Documentation**7.77.2.1 NETGRAPH_INIT (atmlc, & ng_atmlc_typestruct)****7.77.2.2 static int ng_atmlc_constructor (node_p node) [static]**

Definition at line 79 of file ng_atmlc.c.

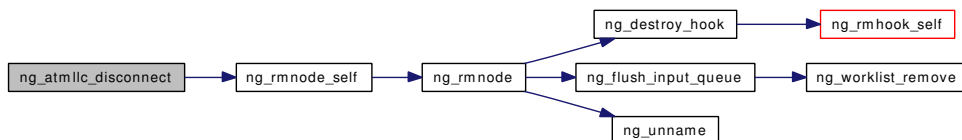
References NG_NODE_SET_PRIVATE.

7.77.2.3 static int ng_atmlc_disconnect (hook_p hook) [static]

Definition at line 253 of file ng_atmlc.c.

References ng_atmlc_priv::atm, ng_atmlc_priv::ether, ng_atmlc_priv::fddi, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:

**7.77.2.4 static int ng_atmlc_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 122 of file ng_atmlc.c.

References ng_atmlc_priv::atm, ng_atmlc_priv::ether, ng_atmlc_priv::fddi, NG_ATMLLC_HOOK_-, NG_ATMLLC_HOOK_ETHER, NG_ATMLLC_HOOK_FDDI, and NG_NODE_PRIVATE.

7.77.2.5 static int ng_atmllc_rcvdata (hook_p hook, item_p item) [static]

Definition at line 151 of file ng_atmllc.c.

References ng_atmllc_priv::atm, ng_atmllc_priv::ether, ng_atmllc_priv::fddi, NG_ATMLLC_HEADER, NG_ATMLLC_HEADER_LEN, NG_ATMLLC_TYPE_ETHERNET_FCS, NG_ATMLLC_TYPE_ETHERNET_NOFCS, NG_ATMLLC_TYPE_FDDI_FCS, NG_ATMLLC_TYPE_FDDI_NOFCS, NG_FREE_ITEM, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_NODE_PRIVATE, and NGI_M.

7.77.2.6 static int ng_atmllc_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 95 of file ng_atmllc.c.

References ng_mesg::ng_msghdr::flags, ng_mesg::header, NG_RESPOND_MSG, NGF_RESP, and NGI_GET_MSG.

7.77.2.7 static int ng_atmllc_shutdown (node_p node) [static]

Definition at line 108 of file ng_atmllc.c.

References NG_NODE_PRIVATE, and NG_NODE_UNREF.

7.77.3 Variable Documentation**7.77.3.1 ng_constructor_t ng_atmllc_constructor [static]**

Definition at line 59 of file ng_atmllc.c.

7.77.3.2 ng_disconnect_t ng_atmllc_disconnect [static]

Definition at line 64 of file ng_atmllc.c.

7.77.3.3 ng_newhook_t ng_atmllc_newhook [static]

Definition at line 62 of file ng_atmllc.c.

7.77.3.4 ng_rcvdata_t ng_atmllc_rcvdata [static]

Definition at line 63 of file ng_atmllc.c.

7.77.3.5 ng_rcvmsg_t ng_atmllc_rcvmsg [static]

Definition at line 61 of file ng_atmllc.c.

7.77.3.6 ng_shutdown_t ng_atmllc_shutdown [static]

Definition at line 60 of file ng_atmllc.c.

7.77.3.7 struct ng_type ng_atmllc_typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_ATMLLC_NODE_TYPE,
    .constructor =  ng_atmllc_constructor,
    .rcvmsg =       ng_atmllc_rcvmsg,
    .shutdown =     ng_atmllc_shutdown,
    .newhook =      ng_atmllc_newhook,
    .rcvdata =      ng_atmllc_rcvdata,
    .disconnect =   ng_atmllc_disconnect,
}
```

Definition at line 66 of file ng_atmllc.c.

7.78 /usr/src/sys/netgraph/ng_atmllc.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_ATMLLC_NODE_TYPE](#) "atmllc"
- #define [NGM_ATMLLC_COOKIE](#) 1065246274
- #define [NG_ATMLLC_HOOK_ATM](#) "atm"
- #define [NG_ATMLLC_HOOK_ETHER](#) "ether"
- #define [NG_ATMLLC_HOOK_802_4](#) "ieee8024"
- #define [NG_ATMLLC_HOOK_802_5](#) "ieee8025"
- #define [NG_ATMLLC_HOOK_802_6](#) "ieee8026"
- #define [NG_ATMLLC_HOOK_FDDI](#) "fddi"
- #define [NG_ATMLLC_HOOK_BPDU](#) "bpdu"

7.78.1 Define Documentation

7.78.1.1 #define NG_ATMLLC_HOOK_802_4 "ieee8024"

Definition at line 39 of file ng_atmllc.h.

7.78.1.2 #define NG_ATMLLC_HOOK_802_5 "ieee8025"

Definition at line 40 of file ng_atmllc.h.

7.78.1.3 #define NG_ATMLLC_HOOK_802_6 "ieee8026"

Definition at line 41 of file ng_atmllc.h.

7.78.1.4 #define NG_ATMLLC_HOOK_ATM "atm"

Definition at line 37 of file ng_atmllc.h.

Referenced by ng_atmllc_newhook().

7.78.1.5 #define NG_ATMLLC_HOOK_BPDU "bpdu"

Definition at line 43 of file ng_atmllc.h.

7.78.1.6 #define NG_ATMLLC_HOOK_ETHER "ether"

Definition at line 38 of file ng_atmllc.h.

Referenced by ng_atmllc_newhook().

7.78.1.7 #define NG_ATMLLC_HOOK_FDDI "fddi"

Definition at line 42 of file ng_atmlc.h.

Referenced by ng_atmlc_newhook().

7.78.1.8 #define NG_ATMLLC_NODE_TYPE "atmlc"

Definition at line 33 of file ng_atmlc.h.

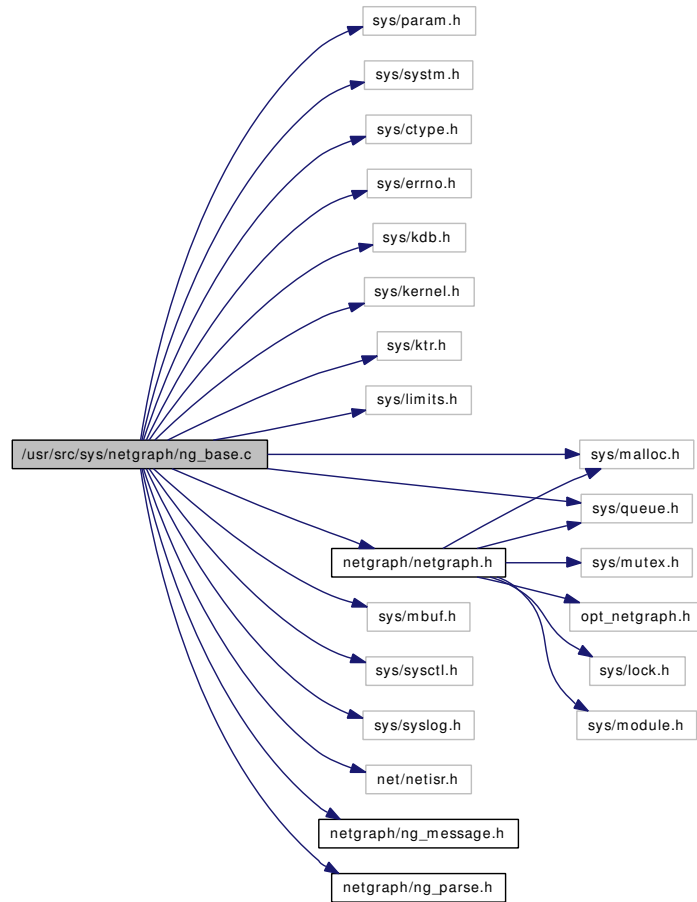
7.78.1.9 #define NGM_ATMLLC_COOKIE 1065246274

Definition at line 34 of file ng_atmlc.h.

7.79 /usr/src/sys/netgraph/ng_base.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/ctype.h>
#include <sys/errno.h>
#include <sys/kdb.h>
#include <sys/kernel.h>
#include <sys/ktr.h>
#include <sys/limits.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <sys/sysctl.h>
#include <sys/syslog.h>
#include <net/netisr.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
```

Include dependency graph for ng_base.c:



Defines

- #define [NG_ID_HASH_SIZE](#) 32
- #define [NG_IDHASH_FN](#)(ID) ((ID) % (NG_ID_HASH_SIZE))
- #define [NG_IDHASH_FIND](#)(ID, node)
- #define [_NG_ALLOC_HOOK](#)(hook) MALLOC(hook, [hook_p](#), sizeof(*hook), M_NETGRAPH_HOOK, M_NOWAIT | M_ZERO)
- #define [_NG_ALLOC_NODE](#)(node) MALLOC(node, [node_p](#), sizeof(*node), M_NETGRAPH_NODE, M_NOWAIT | M_ZERO)
- #define [NG_ALLOC_HOOK](#)(hook) [_NG_ALLOC_HOOK](#)(hook)
- #define [NG_ALLOC_NODE](#)(node) [_NG_ALLOC_NODE](#)(node)
- #define [NG_FREE_HOOK](#)(hook) do { FREE((hook), M_NETGRAPH_HOOK); } while (0)
- #define [NG_FREE_NODE](#)(node) do { FREE((node), M_NETGRAPH_NODE); } while (0)
- #define [TRAP_ERROR](#)()
- #define [CHECK_DATA_MBUF](#)(m)
- #define [DEFINE_PARSE_STRUCT_TYPE](#)(lo, up, args)
- #define [WRITER_ACTIVE](#) 0x00000001
- #define [OP_PENDING](#) 0x00000002
- #define [READER_INCREMENT](#) 0x00000004
- #define [READER_MASK](#) 0xffffffffc
- #define [SAFETY_BARRIER](#) 0x00100000

- #define [NGQ_RMASK](#) (WRITER_ACTIVE|OP_PENDING)
- #define [NGQ_WMASK](#) (NGQ_RMASK|READER_MASK)
- #define [QUEUE_ACTIVE](#)(QP) ((QP) → q_flags & OP_PENDING)
- #define [HEAD_IS_READER](#)(QP) NGI_QUEUED_READER((QP) → queue)
- #define [HEAD_IS_WRITER](#)(QP) NGI_QUEUED_WRITER((QP) → queue)
- #define [QUEUED_READER_CAN_PROCEED](#)(QP) (((QP) → q_flags & (NGQ_RMASK & ~OP_PENDING)) == 0)
- #define [QUEUED_WRITER_CAN_PROCEED](#)(QP) (((QP) → q_flags & (NGQ_WMASK & ~OP_PENDING)) == 0)
- #define [NEXT_QUEUED_ITEM_CAN_PROCEED](#)(QP)
- #define [NGQRW_R](#) 0
- #define [NGQRW_W](#) 1
- #define [ITEM_DEBUG_CHECKS](#)
- #define [SET_RETADDR](#)(item, here, retaddr)
- #define [TESTING](#)

Functions

- [MODULE_VERSION](#) (netgraph, NG_ABI_VERSION)
- static [LIST_HEAD](#) (ng_node)
- static [TAILQ_HEAD](#) (ng_node)
- static int [ng_generic_linkinfo_getLength](#) (const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)
- [DEFINE_PARSE_STRUCT_TYPE](#) (typelist, TYPELIST,(&ng_generic_nodeinfoarray_type))
- [DEFINE_PARSE_STRUCT_TYPE](#) (hooklist, HOOKLIST,(&ng_generic_nodeinfo_type,&ng_generic_linkinfo_array_type))
- [DEFINE_PARSE_STRUCT_TYPE](#) (listnodes, LISTNODES,(&ng_generic_nodeinfoarray_type))
- int [ng_make_node](#) (const char *typename, [node_p](#) *nodepp)
- int [ng_make_node_common](#) (struct [ng_type](#) *type, [node_p](#) *nodepp)
- void [ng_rmnode](#) ([node_p](#) node, [hook_p](#) dummy1, void *dummy2, int dummy3)
- int [ng_unref_node](#) ([node_p](#) node)
- static [node_p](#) [ng_ID2noderef](#) (ng_ID_t ID)
- ng_ID_t [ng_node2ID](#) ([node_p](#) node)
- int [ng_name_node](#) ([node_p](#) node, const char *name)
- [node_p](#) [ng_name2noderef](#) ([node_p](#) here, const char *name)
- static ng_ID_t [ng_decodeidname](#) (const char *name)
- void [ng_unname](#) ([node_p](#) node)
- void [ng_unref_hook](#) ([hook_p](#) hook)
- static int [ng_add_hook](#) ([node_p](#) node, const char *name, [hook_p](#) *hookp)
- [hook_p](#) [ng_findhook](#) ([node_p](#) node, const char *name)
- void [ng_destroy_hook](#) ([hook_p](#) hook)
- int [ng_bypass](#) ([hook_p](#) hook1, [hook_p](#) hook2)
- int [ng_newtype](#) (struct [ng_type](#) *tp)
- int [ng_rmtime](#) (struct [ng_type](#) *tp)
- [ng_type](#) * [ng_findtype](#) (const char *typename)
- static void [ng_con_part3](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static void [ng_con_part2](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static int [ng_con_nodes](#) ([node_p](#) node, const char *name, [node_p](#) node2, const char *name2)
- static int [ng_mkpeer](#) ([node_p](#) node, const char *name, const char *name2, char *type)
- int [ng_rmnode_self](#) ([node_p](#) node)

- static void [ng_rmhook_part2](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- int [ng_rmhook_self](#) ([hook_p](#) hook)
- int [ng_path_parse](#) (char *addr, char **nodep, char **pathp, char **hookp)
- int [ng_path2noderef](#) ([node_p](#) here, const char *address, [node_p](#) *destp, [hook_p](#) *lasthook)
- static __inline [item_p](#) [ng_dequeue](#) (struct [ng_queue](#) *ngq, int *rw)
- static __inline [item_p](#) [ng_acquire_read](#) (struct [ng_queue](#) *ngq, [item_p](#) item)
- static __inline [item_p](#) [ng_acquire_write](#) (struct [ng_queue](#) *ngq, [item_p](#) item)
- static __inline void [ng_leave_read](#) (struct [ng_queue](#) *ngq)
- static __inline void [ng_leave_write](#) (struct [ng_queue](#) *ngq)
- static __inline void [ng_queue_rw](#) (struct [ng_queue](#) *ngq, [item_p](#) item, int rw)
- static void [ng_flush_input_queue](#) (struct [ng_queue](#) *ngq)
- int [ng_snd_item](#) ([item_p](#) item, int flags)
- static int [ng_apply_item](#) ([node_p](#) node, [item_p](#) item, int rw)
- static int [ng_generic_msg](#) ([node_p](#) here, [item_p](#) item, [hook_p](#) lasthook)
- [TUNABLE_INT](#) ("net.graph.maxalloc",&maxalloc)
- [SYSCTL_INT](#) (_net_graph, OID_AUTO, maxalloc, CTLFLAG_RD, 0, "Maximum number of queue items to allocate")
- static __inline [item_p](#) [ng_getqblk](#) (int flags)
- void [ng_free_item](#) ([item_p](#) item)
- int [ng_mod_event](#) (module_t mod, int event, void *data)
- static int [ngb_mod_event](#) (module_t mod, int event, void *data)
- [DECLARE_MODULE](#) (netgraph, [netgraph_mod](#), SI_SUB_NETGRAPH, SI_ORDER_MIDDLE)
- [SYSCTL_NODE](#) (_net, OID_AUTO, graph, CTLFLAG_RW, 0, "netgraph Family")
- [SYSCTL_INT](#) (_net_graph, OID_AUTO, abi_version, CTLFLAG_RD, 0, NG_ABI_VERSION, "")
- [SYSCTL_INT](#) (_net_graph, OID_AUTO, msg_version, CTLFLAG_RD, 0, NG_VERSION, "")
- static void [ngintr](#) (void)
- static void [ng_worklist_remove](#) ([node_p](#) node)
- static void [ng_setisr](#) ([node_p](#) node)
- [item_p](#) [ng_package_data](#) (struct mbuf *m, int flags)
- [item_p](#) [ng_package_msg](#) (struct [ng_mesg](#) *msg, int flags)
- int [ng_address_hook](#) ([node_p](#) here, [item_p](#) item, [hook_p](#) hook, [ng_ID_t](#) retaddr)
- int [ng_address_path](#) ([node_p](#) here, [item_p](#) item, char *address, [ng_ID_t](#) retaddr)
- int [ng_address_ID](#) ([node_p](#) here, [item_p](#) item, [ng_ID_t](#) ID, [ng_ID_t](#) retaddr)
- [item_p](#) [ng_package_msg_self](#) ([node_p](#) here, [hook_p](#) hook, struct [ng_mesg](#) *msg)
- int [ng_send_fn1](#) ([node_p](#) node, [hook_p](#) hook, [ng_item_fn](#) *fn, void *arg1, int arg2, int flags)
- static void [ng_callout_trampoline](#) (void *arg)
- int [ng_callout](#) (struct callout *c, [node_p](#) node, [hook_p](#) hook, int ticks, [ng_item_fn](#) *fn, void *arg1, int arg2)
- int [ng_uncallout](#) (struct callout *c, [node_p](#) node)
- void [ng_replace_retaddr](#) ([node_p](#) here, [item_p](#) item, [ng_ID_t](#) retaddr)
- void [ng_macro_test](#) ([item_p](#) item)

Variables

- [ng_node](#) [ng_deadnode](#)
- [ng_hook](#) [ng_deadhook](#)
- static struct [ng_parse_array_info](#) [ng_nodeinfoarray_type_info](#)
- static struct [ng_parse_type](#) [ng_generic_nodeinfoarray_type](#)
- static struct [ng_parse_array_info](#) [ng_typeinfoarray_type_info](#)
- static struct [ng_parse_type](#) [ng_generic_typeinfoarray_type](#)

- static struct [ng_parse_array_info](#) [ng_generic_linkinfo_array_type_info](#)
- static struct [ng_parse_type](#) [ng_generic_linkinfo_array_type](#)
- static struct [ng_cmdlist](#) [ng_generic_cmds](#) []
- [uma_zone_t](#) [ng_qzone](#)
- static int [maxalloc](#) = 512
- static moduledata_t [netgraph_mod](#)

7.79.1 Define Documentation

7.79.1.1 `#define _NG_ALLOC_HOOK(hook) MALLOC(hook, hook_p, sizeof(*hook), M_NETGRAPH_HOOK, M_NOWAIT | M_ZERO)`

7.79.1.2 `#define _NG_ALLOC_NODE(node) MALLOC(node, node_p, sizeof(*node), M_NETGRAPH_NODE, M_NOWAIT | M_ZERO)`

7.79.1.3 `#define CHECK_DATA_MBUF(m)`

Referenced by [ng_snd_item\(\)](#).

7.79.1.4 `#define DEFINE_PARSE_STRUCT_TYPE(lo, up, args)`

Value:

```
static const struct ng_parse_struct_field          \
    ng_ ## lo ## _type_fields[] = NG_GENERIC_ ## up ## _INFO args; \
static const struct ng_parse_type ng_generic_ ## lo ## _type = { \
    &ng_parse_struct_type, \
    &ng_ ## lo ## _type_fields \
}
```

7.79.1.5 `#define HEAD_IS_READER(QP) NGL_QUEUED_READER((QP) → queue)`

Definition at line 1786 of file [ng_base.c](#).

Referenced by [ng_dequeue\(\)](#).

7.79.1.6 `#define HEAD_IS_WRITER(QP) NGL_QUEUED_WRITER((QP) → queue)`

Definition at line 1787 of file [ng_base.c](#).

7.79.1.7 `#define ITEM_DEBUG_CHECKS`

Definition at line 3356 of file [ng_base.c](#).

Referenced by [ng_address_hook\(\)](#), [ng_package_data\(\)](#), and [ng_package_msg\(\)](#).

7.79.1.8 `#define NEXT_QUEUED_ITEM_CAN_PROCEED(QP)`

Value:

```
(QUEUE_ACTIVE(QP) &&
 ((HEAD_IS_READER(QP)) ? QUEUED_READER_CAN_PROCEED(QP) :
  QUEUED_WRITER_CAN_PROCEED(QP)))
```

Definition at line 1796 of file ng_base.c.

Referenced by ng_dequeue(), ng_queue_rw(), and ng_snd_item().

7.79.1.9 #define NG_ALLOC_HOOK(hook) _NG_ALLOC_HOOK(hook)

Referenced by ng_add_hook(), and ng_con_nodes().

7.79.1.10 #define NG_ALLOC_NODE(node) _NG_ALLOC_NODE(node)

Referenced by ng_make_node_common().

7.79.1.11 #define NG_FREE_HOOK(hook) do { FREE((hook), M_NETGRAPH_HOOK); } while (0)

Referenced by ng_unref_hook().

7.79.1.12 #define NG_FREE_NODE(node) do { FREE((node), M_NETGRAPH_NODE); } while (0)

Referenced by ng_unref_node().

7.79.1.13 #define NG_ID_HASH_SIZE 32

7.79.1.14 #define NG_IDHASH_FIND(ID, node)

Value:

```
do {
    mtx_assert(&ng_idhash_mtx, MA_OWNED);
    LIST_FOREACH(node, &ng_id_hash[NG_IDHASH_FN(ID)],
                 nd_idnodes) {
        if (NG_NODE_IS_VALID(node)
            && (NG_NODE_ID(node) == ID)) {
            break;
        }
    }
} while (0)
```

Referenced by ng_ID2noderef(), and ng_make_node_common().

7.79.1.15 #define NG_IDHASH_FN(ID) ((ID) % (NG_ID_HASH_SIZE))

Referenced by ng_make_node_common().

7.79.1.16 #define NGQ_RMASK (WRITER_ACTIVE|OP_PENDING)

Definition at line 1777 of file ng_base.c.

Referenced by ng_acquire_read().

7.79.1.17 #define NGQ_WMASK (NGQ_RMASK|READER_MASK)

Definition at line 1780 of file ng_base.c.

Referenced by ng_acquire_write().

7.79.1.18 #define NGQRW_R 0

Definition at line 1802 of file ng_base.c.

Referenced by ng_acquire_read(), ng_apply_item(), ng_dequeue(), and ng_snd_item().

7.79.1.19 #define NGQRW_W 1

Definition at line 1803 of file ng_base.c.

Referenced by ng_acquire_write(), ng_dequeue(), ng_queue_rw(), and ng_snd_item().

7.79.1.20 #define OP_PENDING 0x00000002

Definition at line 1770 of file ng_base.c.

Referenced by ng_dequeue(), ng_flush_input_queue(), and ng_queue_rw().

7.79.1.21 #define QUEUE_ACTIVE(QP) ((QP) → q_flags & OP_PENDING)

Definition at line 1783 of file ng_base.c.

Referenced by ng_dequeue().

7.79.1.22 #define QUEUED_READER_CAN_PROCEED(QP) (((QP) → q_flags & (NGQ_RMASK & ~OP_PENDING)) == 0)

Definition at line 1790 of file ng_base.c.

Referenced by ng_dequeue().

7.79.1.23 #define QUEUED_WRITER_CAN_PROCEED(QP) (((QP) → q_flags & (NGQ_WMASK & ~OP_PENDING)) == 0)

Definition at line 1792 of file ng_base.c.

Referenced by ng_dequeue().

7.79.1.24 #define READER_INCREMENT 0x00000004

Definition at line 1771 of file ng_base.c.

Referenced by ng_acquire_read(), ng_dequeue(), and ng_leave_read().

7.79.1.25 #define READER_MASK 0xffffffc

Definition at line 1772 of file ng_base.c.

Referenced by ng_acquire_write().

7.79.1.26 #define SAFETY_BARRIER 0x00100000

Definition at line 1773 of file ng_base.c.

7.79.1.27 #define SET_RETADDR(item, here, retaddr)**Value:**

```
do { /* Data or fn items don't have retaddrs */
    if ((item->el_flags & NGQF_TYPE) == NGQF_MESG) {
        if (retaddr) {
            NGI_RETADDR(item) = retaddr;
        } else {
            /*
             * The old return address should be ok.
             * If there isn't one, use the address
             * here.
             */
            if (NGI_RETADDR(item) == 0) {
                NGI_RETADDR(item)
                    = ng_node2ID(here);
            }
        }
    }
} while (0)
```

Definition at line 3421 of file ng_base.c.

Referenced by ng_address_hook().

7.79.1.28 #define TESTING

Definition at line 3657 of file ng_base.c.

7.79.1.29 #define TRAP_ERROR()

Referenced by ng_add_hook(), ng_address_hook(), ng_apply_item(), ng_bind(), ng_bypass(), ng_con_nodes(), ng_con_part2(), ng_generic_msg(), ng_make_node(), ng_make_node_common(), ng_name_node(), ng_newtype(), ng_path2noderef(), ng_rmtype(), ng_snd_item(), ngs_rcvdata(), and ngs_rcvmsg().

7.79.1.30 #define WRITER_ACTIVE 0x00000001

Definition at line 1769 of file ng_base.c.

Referenced by ng_acquire_write(), ng_dequeue(), and ng_leave_write().

7.79.2 Function Documentation**7.79.2.1 DECLARE_MODULE (netgraph, netgraph_mod, SI_SUB_NETGRAPH, SI_ORDER_MIDDLE)****7.79.2.2 DEFINE_PARSE_STRUCT_TYPE (listnodes, LISTNODES, & ng_generic_nodeinfoarray_type)****7.79.2.3 DEFINE_PARSE_STRUCT_TYPE (hooklist, HOOKLIST, && ng_generic_linkinfo_array_type)****7.79.2.4 DEFINE_PARSE_STRUCT_TYPE (typelist, TYPELIST, & ng_generic_nodeinfoarray_type)****7.79.2.5 static LIST_HEAD (ng_node) [static]**

Definition at line 72 of file ng_base.c.

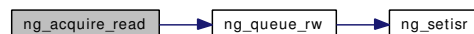
7.79.2.6 MODULE_VERSION (netgraph, NG_ABI_VERSION)**7.79.2.7 static __inline item_p ng_acquire_read (struct ng_queue * ngq, item_p item) [static]**

Definition at line 2025 of file ng_base.c.

References ng_node::nd_ID, ng_node::nd_input_queue, ng_deadnode, ng_queue_rw(), NGQ_RMASK, NGQRW_R, ng_queue::q_flags, ng_queue::q_mtx, ng_queue::q_node, and READER_INCREMENT.

Referenced by ng_snd_item().

Here is the call graph for this function:

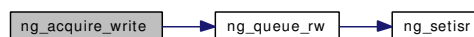
**7.79.2.8 static __inline item_p ng_acquire_write (struct ng_queue * ngq, item_p item) [static]**

Definition at line 2069 of file ng_base.c.

References ng_node::nd_ID, ng_node::nd_input_queue, ng_deadnode, ng_queue_rw(), NGQ_WMASK, NGQRW_W, ng_queue::q_flags, ng_queue::q_mtx, ng_queue::q_node, READER_MASK, and WRITER_ACTIVE.

Referenced by ng_snd_item().

Here is the call graph for this function:



7.79.2.9 `static int ng_add_hook (node_p node, const char * name, hook_p * hookp)` [static]

Definition at line 947 of file `ng_base.c`.

References `ng_hook::hk_flags`, `HK_INVALID`, `ng_hook::hk_node`, `ng_hook::hk_peer`, `ng_hook::hk_refs`, `ng_node::nd_numhooks`, `ng_node::nd_type`, `ng_type::newhook`, `NG_ALLOC_HOOK`, `ng_deadhook`, `ng_findhook()`, `NG_HOOK_NAME`, `NG_HOOK_REF`, `NG_HOOK_UNREF`, `NG_HOOKSIZ`, `NG_NODE_REF`, and `TRAP_ERROR`.

Referenced by `ng_con_nodes()`, and `ng_mkpeer()`.

Here is the call graph for this function:



7.79.2.10 `int ng_address_hook (node_p here, item_p item, hook_p hook, ng_ID_t retaddr)`

Definition at line 3437 of file `ng_base.c`.

References `ITEM_DEBUG_CHECKS`, `NG_FREE_ITEM`, `NG_HOOK_NOT_VALID`, `NG_HOOK_PEER`, `NG_HOOK_REF`, `NG_NODE_NOT_VALID`, `NG_NODE_REF`, `NG_PEER_NODE`, `NGI_SET_HOOK`, `NGI_SET_NODE`, `SET_RETADDR`, and `TRAP_ERROR`.

7.79.2.11 `int ng_address_ID (node_p here, item_p item, ng_ID_t ID, ng_ID_t retaddr)`

Definition at line 3498 of file `ng_base.c`.

7.79.2.12 `int ng_address_path (node_p here, item_p item, char * address, ng_ID_t retaddr)`

Definition at line 3472 of file `ng_base.c`.

Referenced by `ng_connect_data()`, and `ngc_send()`.

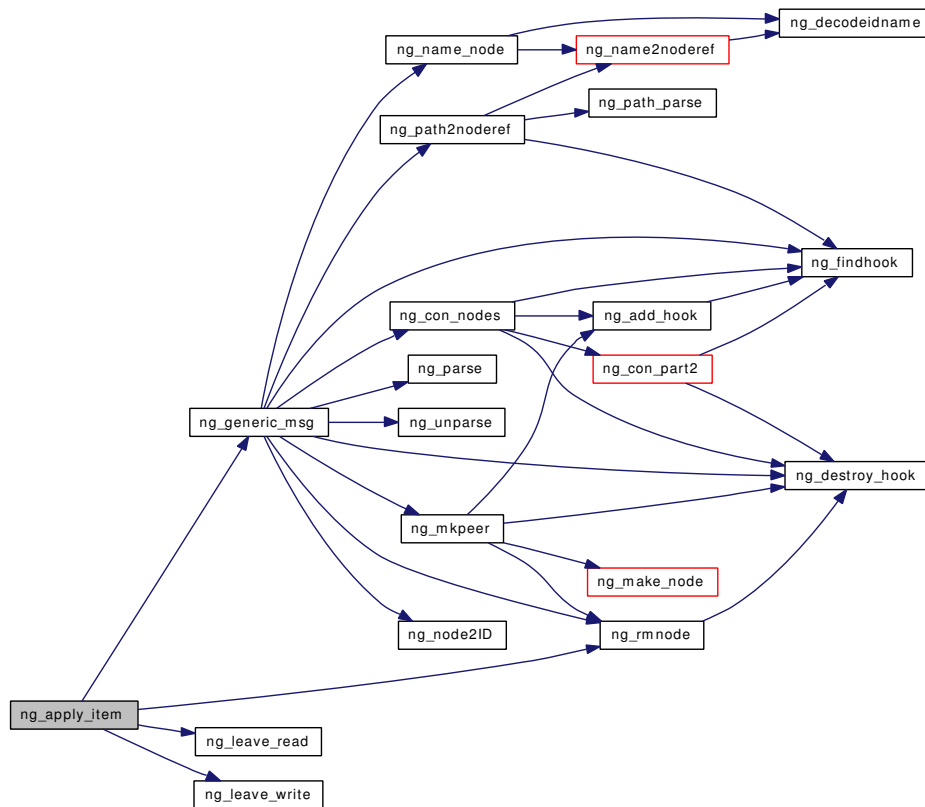
7.79.2.13 `static int ng_apply_item (node_p node, item_p item, int rw)` [static]

Definition at line 2326 of file `ng_base.c`.

References `ng_item::apply`, `ng_item::context`, `ng_item::el_flags`, `ng_msg::ng_msghdr::flags`, `ng_msg::header`, `ng_hook::hk_rcvdata`, `ng_node::nd_input_queue`, `ng_node::nd_type`, `NG_FREE_ITEM`, `ng_generic_msg()`, `NG_HOOK_NODE`, `NG_HOOK_NOT_VALID`, `NG_HOOK_UNREF`, `ng_leave_read()`, `ng_leave_write()`, `NG_NODE_NOT_VALID`, `ng_rmnode()`, `NGF_RESP`, `NGI_ARG1`, `NGI_ARG2`, `NGI_FN`, `NGI_GET_HOOK`, `NGI_MSG`, `NGM_GENERIC_COOKIE`, `NGQF_DATA`, `NGQF_FN`, `NGQF_MSG`, `NGQF_TYPE`, `NGQRW_R`, `ng_type::rcvmsg`, `TRAP_ERROR`, and `ng_msg::ng_msghdr::typecookie`.

Referenced by `ng_snd_item()`, and `ngintr()`.

Here is the call graph for this function:



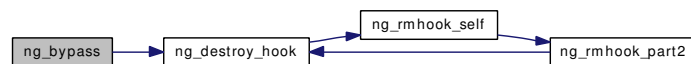
7.79.2.14 `int ng_bypass (hook_p hook1, hook_p hook2)`

Definition at line 1120 of file `ng_base.c`.

References `ng_hook::hk_node`, `ng_hook::hk_peer`, `ng_deadhook`, `ng_destroy_hook()`, `NG_HOOK_UNREF`, and `TRAP_ERROR`.

Referenced by `ngt_close()`.

Here is the call graph for this function:



7.79.2.15 `int ng_callout (struct callout * c, node_p node, hook_p hook, int ticks, ng_item_fn * fn, void * arg1, int arg2)`

Definition at line 3588 of file `ng_base.c`.

Referenced by `cisco_keepalive()`, `cisco_newhook()`, `LMI_ticker()`, `ng_bridge_constructor()`, `ng_h4_timeout()`, `ng_hci_command_timeout()`, `ng_hci_con_timeout()`, `ng_l2cap_command_timeout()`, `ng_l2cap_discon_timeout()`, `ng_l2cap_lp_timeout()`, `ng_l2tp_rcv_ctrl()`, `ng_l2tp_seq_rack_timeout()`, `ng_`

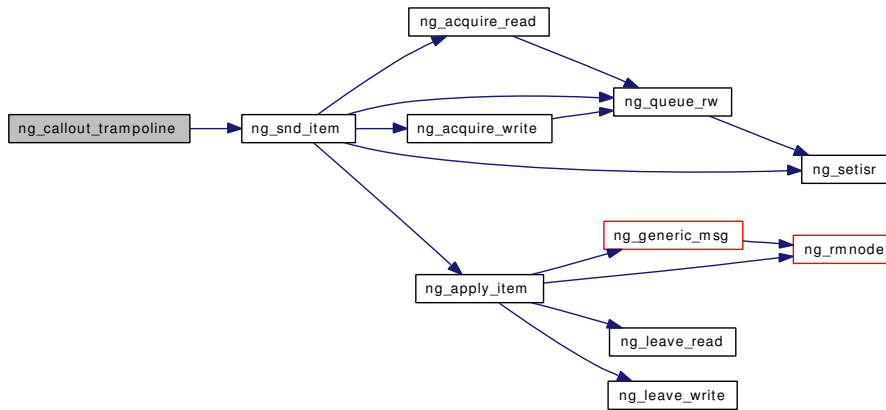
l2tp_seq_rcv_ns(), ng_ppp_start_frag_timer(), ng_pppoe_sendpacket(), ng_pptpgre_start_rcv_ack_timer(), ng_pptpgre_start_send_ack_timer(), ng_source_intr(), ng_source_start(), nglmi_startup(), ngt_start(), and pppoe_ticker().

7.79.2.16 static void ng_callout_trampoline (void * arg) [static]

Definition at line 3579 of file ng_base.c.

References ng_snd_item().

Here is the call graph for this function:



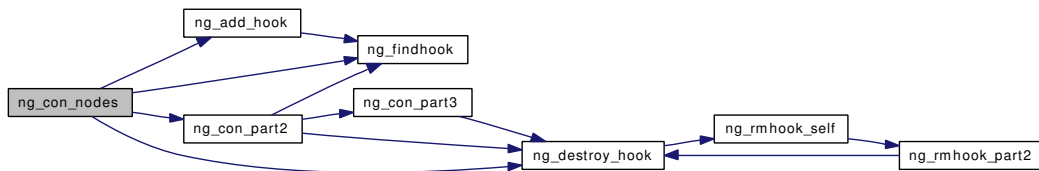
7.79.2.17 static int ng_con_nodes (node_p node, const char * name, node_p node2, const char * name2) [static]

Definition at line 1347 of file ng_base.c.

References ng_hook::hk_flags, HK_INVALID, ng_hook::hk_node, ng_hook::hk_peer, ng_hook::hk_refs, ng_add_hook(), NG_ALLOC_HOOK, ng_con_part2(), ng_deadnode, ng_destroy_hook(), ng_findhook(), NG_HOOK_NAME, NG_HOOK_REF, NG_HOOK_UNREF, NG_HOOKSIZ, ng_send_fn, and TRAP_ERROR.

Referenced by ng_generic_msg().

Here is the call graph for this function:



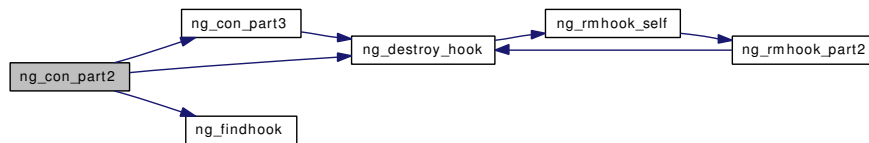
7.79.2.18 static void ng_con_part2 (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 1259 of file ng_base.c.

References `ng_type::connect`, `ng_hook::hk_flags`, `HK_INVALID`, `ng_hook::hk_name`, `ng_hook::hk_node`, `ng_hook::hk_peer`, `ng_node::nd_numhooks`, `ng_node::nd_type`, `ng_type::newhook`, `ng_con_part3()`, `ng_deadhook`, `ng_destroy_hook()`, `ng_findhook()`, `NG_HOOK_NAME`, `NG_HOOK_REF`, `NG_NODE_REF`, `ng_send_fn`, and `TRAP_ERROR`.

Referenced by `ng_con_nodes()`.

Here is the call graph for this function:



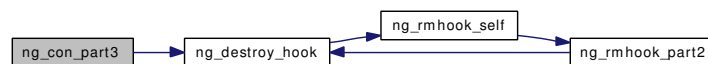
7.79.2.19 `static void ng_con_part3 (node_p node, hook_p hook, void * arg1, int arg2)` [static]

Definition at line 1219 of file `ng_base.c`.

References `ng_type::connect`, `ng_hook::hk_flags`, `HK_INVALID`, `ng_hook::hk_node`, `ng_node::nd_type`, `ng_deadnode`, `ng_destroy_hook()`, and `NG_HOOK_NODE`.

Referenced by `ng_con_part2()`.

Here is the call graph for this function:



7.79.2.20 `static ng_ID_t ng_decodeidname (const char * name)` [static]

Definition at line 877 of file `ng_base.c`.

References `ng_ID_t`.

Referenced by `ng_name2noderef()`, and `ng_name_node()`.

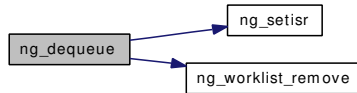
7.79.2.21 `static __inline item_p ng_dequeue (struct ng_queue * ngq, int * rw)` [static]

Definition at line 1815 of file `ng_base.c`.

References `ng_item::el_flags`, `ng_item::el_next`, `HEAD_IS_READER`, `ng_queue::last`, `ng_node::nd_ID`, `NEXT_QUEUED_ITEM_CAN_PROCEED`, `ng_setisr()`, `ng_worklist_remove()`, `NGQRW_R`, `NGQRW_W`, `OP_PENDING`, `ng_queue::q_flags`, `ng_queue::q_mtx`, `ng_queue::q_node`, `ng_queue::queue`, `QUEUE_ACTIVE`, `QUEUED_READER_CAN_PROCEED`, `QUEUED_WRITER_CAN_PROCEED`, `READER_INCREMENT`, and `WRITER_ACTIVE`.

Referenced by `ngintr()`.

Here is the call graph for this function:



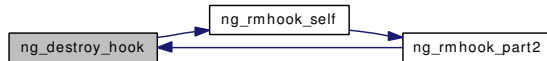
7.79.2.22 void ng_destroy_hook (hook_p hook)

Definition at line 1041 of file ng_base.c.

References `_NG_HOOK_NODE`, `ng_type::disconnect`, `ng_hook::hk_flags`, `HK_INVALID`, `ng_hook::hk_peer`, `ng_node::nd_numhooks`, `ng_node::nd_type`, `ng_deadhook`, `ng_deadnode`, `NG_HOOK_NODE`, `NG_HOOK_PEER`, `NG_HOOK_UNREF`, `NG_NODE_UNREF`, and `ng_rmhook_self()`.

Referenced by `ng_bypass()`, `ng_con_nodes()`, `ng_con_part2()`, `ng_con_part3()`, `ng_generic_msg()`, `ng_mkpeer()`, `ng_rmhook_part2()`, and `ng_rmnode()`.

Here is the call graph for this function:



7.79.2.23 hook_p ng_findhook (node_p node, const char * name)

Definition at line 1008 of file ng_base.c.

References `ng_type::findhook`, `ng_node::nd_type`, `NG_HOOK_IS_VALID`, and `NG_HOOK_NAME`.

Referenced by `ng_add_hook()`, `ng_bpf_rcvdata()`, `ng_bpf_rcvmsg()`, `ng_con_nodes()`, `ng_con_part2()`, `ng_etf_rcvmsg()`, `ng_generic_msg()`, `ng_path2noderef()`, `ng_tag_rcvmsg()`, `ng_tag_setdata_in()`, `ng_tcpmss_rcvmsg()`, `ng_vlan_rcvmsg()`, `ngd_send()`, and `ngh_rcvmsg()`.

7.79.2.24 struct ng_type* ng_findtype (const char * typename)

Definition at line 1199 of file ng_base.c.

References `ng_type::name`.

Referenced by `ng_make_node()`, `ng_make_node_common()`, `ng_newtype()`, and `ngc_send()`.

7.79.2.25 static void ng_flush_input_queue (struct ng_queue * ngq) [static]

Definition at line 2117 of file ng_base.c.

References `ng_item::apply`, `ng_item::context`, `ng_item::el_next`, `ng_queue::last`, `NG_FREE_ITEM`, `ng_worklist_remove()`, `OP_PENDING`, `ng_queue::q_flags`, `ng_queue::q_mtx`, `ng_queue::q_node`, and `ng_queue::queue`.

Referenced by `ng_rmnode()`.

Here is the call graph for this function:



7.79.2.26 void ng_free_item (item_p item)

Definition at line 2947 of file ng_base.c.

References `_NGI_ARG1`, `_NGI_ARG2`, `_NGI_CLR_HOOK`, `_NGI_CLR_NODE`, `_NGI_FN`, `_NGI_M`, `_NGI_MSG`, `_NGI_RETADDR`, `ng_item::apply`, `ng_item::el_flags`, `NG_FREE_M`, `NG_FREE_MSG`, `NGQF_DATA`, `NGQF_FN`, `NGQF_MESG`, `NGQF_TYPE`, and `NGQF_UNDEF`.

7.79.2.27 static int ng_generic_linkinfo_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]

Definition at line 386 of file ng_base.c.

References `nodeinfo::hooks`, and `hooklist::nodeinfo`.

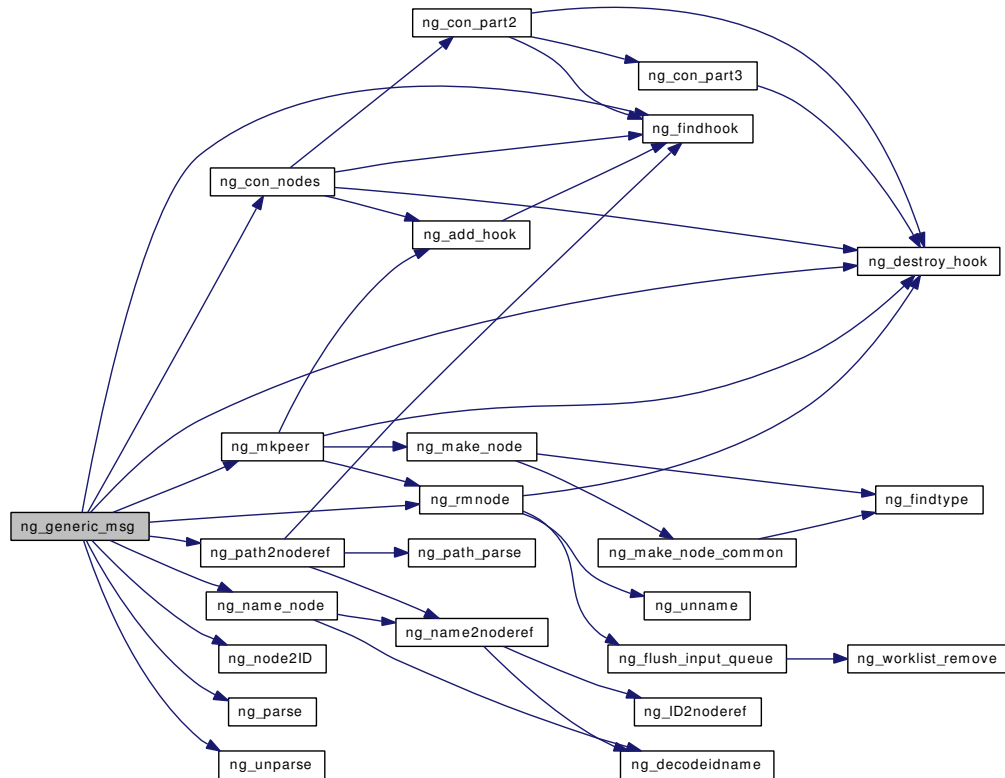
7.79.2.28 static int ng_generic_msg (node_p here, item_p item, hook_p lasthook) [static]

Definition at line 2478 of file ng_base.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_cmdlist::cmd`, `ng_mesg::ng_msghdr::cmd`, `ng_type::cmdlist`, `ng_cmdlist::cookie`, `ng_mesg::data`, `ng_mesg::header`, `nodeinfo::hooks`, `nodeinfo::id`, `ng_cmdlist::name`, `nodeinfo::name`, `ng_type::name`, `ng_node::nd_numhooks`, `ng_node::nd_type`, `ng_con_nodes()`, `ng_destroy_hook()`, `ng_findhook()`, `NG_FREE_MSG`, `ng_generic_cmds`, `NG_HOOK_NAME`, `NG_HOOK_NOT_VALID`, `ng_mkpeer()`, `NG_MKRESPONSE`, `ng_name_node()`, `ng_node2ID()`, `NG_NODE_HAS_NAME`, `NG_NODE_IS_VALID`, `NG_NODE_NAME`, `NG_NODE_NOT_VALID`, `NG_NODE_UNREF`, `ng_parse()`, `ng_path2noderef()`, `NG_PEER_HOOK_NAME`, `NG_PEER_NODE`, `NG_PEER_NODE_NAME`, `ng_rmnode()`, `ng_unparse()`, `NGF_RESP`, `NGI_GET_MSG`, `NGM_ASCII2BINARY`, `NGM_BINARY2ASCII`, `NGM_CONNECT`, `NGM_GENERIC_COOKIE`, `NGM_LISTHOOKS`, `NGM_LISTNAMES`, `NGM_LISTNODES`, `NGM_LISTTYPES`, `NGM_MKPEER`, `NGM_NAME`, `NGM_NODEINFO`, `NGM_RMHOOK`, `NGM_SHUTDOWN`, `linkinfo::nodeinfo`, `typeinfo::numnodes`, `linkinfo::ourhook`, `linkinfo::peerhook`, `ng_type::refs`, `TRAP_ERROR`, `nodeinfo::type`, `typeinfo::type_name`, and `ng_mesg::ng_msghdr::typecookie`.

Referenced by `ng_apply_item()`.

Here is the call graph for this function:



7.79.2.29 `static __inline item_p ng_getqblk(int flags)` [static]

Definition at line 2922 of file `ng_base.c`.

References `NG_WAITOK`.

Referenced by `ng_package_data()`, `ng_package_msg()`, and `ng_package_msg_self()`.

7.79.2.30 `static node_p ng_ID2noderef(ng_ID_t ID)` [static]

Definition at line 774 of file `ng_base.c`.

References `NG_IDHASH_FIND`, and `NG_NODE_REF`.

Referenced by `ng_name2noderef()`.

7.79.2.31 `static __inline void ng_leave_read(struct ng_queue *ngq)` [static]

Definition at line 2105 of file `ng_base.c`.

References `ng_queue::q_flags`, and `READER_INCREMENT`.

Referenced by `ng_apply_item()`.

7.79.2.32 `static __inline void ng_leave_write (struct ng_queue * ngq)` [static]

Definition at line 2111 of file ng_base.c.

References `ng_queue::q_flags`, and `WRITER_ACTIVE`.

Referenced by `ng_apply_item()`.

7.79.2.33 `void ng_macro_test (item_p item)`

Definition at line 3663 of file ng_base.c.

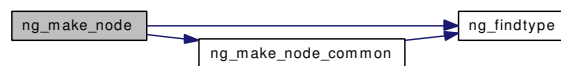
7.79.2.34 `int ng_make_node (const char * typename, node_p * nodepp)`

Definition at line 541 of file ng_base.c.

References `ng_type::constructor`, `ng_findtype()`, `ng_make_node_common()`, `NG_NODE_UNREF`, and `TRAP_ERROR`.

Referenced by `ng_mkpeer()`.

Here is the call graph for this function:

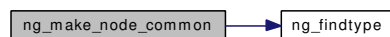
**7.79.2.35** `int ng_make_node_common (struct ng_type * type, node_p * nodepp)`

Definition at line 588 of file ng_base.c.

References `ng_queue::last`, `ng_type::name`, `ng_node::nd_ID`, `ng_node::nd_input_queue`, `ng_node::nd_type`, `NG_ALLOC_NODE`, `ng_findtype()`, `NG_IDHASH_FIND`, `NG_IDHASH_FN`, `NG_NODE_REF`, `ng_queue::q_flags`, `ng_queue::q_mtx`, `ng_queue::q_node`, `ng_queue::queue`, `ng_type::refs`, and `TRAP_ERROR`.

Referenced by `bt3c_pccard_attach()`, `ng_atm_attach()`, `ng_attach_cntl()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_ether_attach()`, `ng_gif_attach()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_ipfw_mod_event()`, `ng_ksocket_finish_accept()`, `ng_make_node()`, `ng_ubt_shutdown()`, and `ngt_open()`.

Here is the call graph for this function:

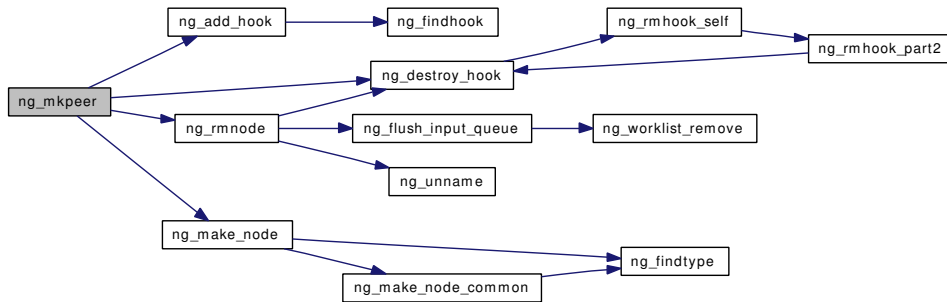
**7.79.2.36** `static int ng_mkpeer (node_p node, const char * name, const char * name2, char * type)` [static]

Definition at line 1405 of file ng_base.c.

References `ng_type::connect`, `ng_hook::hk_flags`, `HK_INVALID`, `ng_hook::hk_node`, `ng_hook::hk_peer`, `ng_node::nd_type`, `ng_add_hook()`, `ng_destroy_hook()`, `NG_HOOK_REF`, `NG_HOOK_UNREF`, `ng_make_node()`, and `ng_rmnode()`.

Referenced by `ng_generic_msg()`.

Here is the call graph for this function:

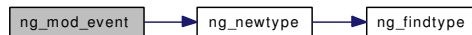


7.79.2.37 `int ng_mod_event (module_t mod, int event, void * data)`

Definition at line 2996 of file `ng_base.c`.

References `ng_type::mod_event`, `ng_newtype()`, and `ng_type::refs`.

Here is the call graph for this function:



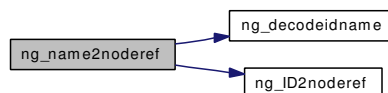
7.79.2.38 `node_p ng_name2noderef (node_p here, const char * name)`

Definition at line 841 of file `ng_base.c`.

References `ng_decodeidname()`, `ng_ID2noderef()`, `ng_ID_t`, `NG_NODE_HAS_NAME`, `NG_NODE_IS_INVALID`, `NG_NODE_NAME`, and `NG_NODE_REF`.

Referenced by `ng_name_node()`, and `ng_path2noderef()`.

Here is the call graph for this function:



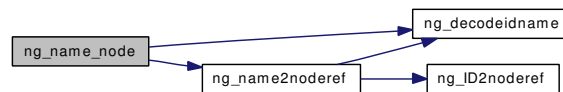
7.79.2.39 `int ng_name_node (node_p node, const char * name)`

Definition at line 799 of file `ng_base.c`.

References `ng_decodeidname()`, `ng_name2noderef()`, `NG_NODE_NAME`, `NG_NODE_UNREF`, `NG_NODESIZ`, and `TRAP_ERROR`.

Referenced by `bt3c_pccard_attach()`, `ng_atm_attach()`, `ng_bind()`, `ng_bt3c_shutdown()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_hci_raw_node_shutdown()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_node_shutdown()`, `ng_btsocket_l2cap_raw_init()`, `ng_btsocket_l2cap_raw_node_shutdown()`, `ng_eiface_constructor()`, `ng_ether_attach()`, `ng_fec_constructor()`, `ng_generic_msg()`, `ng_gif_attach()`, `ng_h4_open()`, `ng_h4_shutdown()`, `ng_iface_constructor()`, `ng_ipfw_mod_event()`, `ng_sppp_constructor()`, `ng_ubt_shutdown()`, and `ngt_open()`.

Here is the call graph for this function:



7.79.2.40 `int ng_newtype (struct ng_type * tp)`

Definition at line 1145 of file `ng_base.c`.

References `ng_type::name`, `NG_ABI_VERSION`, `ng_findtype()`, `NG_TYPESIZ`, `ng_type::refs`, `TRAP_ERROR`, and `ng_type::version`.

Referenced by `bt3c_modevent()`, `ng_btsocket_hci_raw_init()`, `ng_btsocket_l2cap_init()`, `ng_btsocket_l2cap_raw_init()`, `ng_mod_event()`, and `ubt_modevent()`.

Here is the call graph for this function:



7.79.2.41 `ng_ID_t ng_node2ID (node_p node)`

Definition at line 786 of file `ng_base.c`.

References `NG_NODE_ID`.

Referenced by `ng_bridge_nodename()`, `ng_generic_msg()`, `ng_h4_ioctl()`, `ng_package_msg_self()`, and `ngt_ioctl()`.

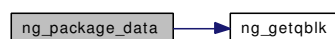
7.79.2.42 `item_p ng_package_data (struct mbuf * m, int flags)`

Definition at line 3373 of file `ng_base.c`.

References `ng_item::el_flags`, `ng_item::el_next`, `ITEM_DEBUG_CHECKS`, `NG_FREE_M`, `ng_getqblk()`, `NGI_M`, `NGQF_DATA`, and `NGQF_READER`.

Referenced by `get_export_dgram()`, `ng_connect_data()`, `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, and `ng_ppp_mp_xmit()`.

Here is the call graph for this function:



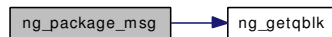
7.79.2.43 `item_p ng_package_msg (struct ng_mesg * msg, int flags)`

Definition at line 3396 of file ng_base.c.

References `ng_mesg::ng_msghdr::cmd`, `ng_item::el_flags`, `ng_item::el_next`, `ng_mesg::header`, `ITEM_DEBUG_CHECKS`, `NG_FREE_MSG`, `ng_getqblk()`, `NGI_MSG`, `NGI_RETADDR`, `NGM_READONLY`, `NGQF_MESG`, `NGQF_READER`, and `NGQF_WRITER`.

Referenced by `ngc_send()`.

Here is the call graph for this function:

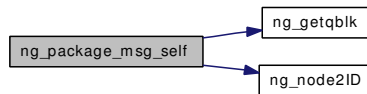


7.79.2.44 `item_p ng_package_msg_self (node_p here, hook_p hook, struct ng_mesg * msg)`

Definition at line 3525 of file ng_base.c.

References `ng_item::el_flags`, `ng_item::el_next`, `NG_FREE_MSG`, `ng_getqblk()`, `NG_HOOK_REF`, `ng_node2ID()`, `NG_NODE_REF`, `NG_NOFLAGS`, `NGI_MSG`, `NGI_RETADDR`, `NGI_SET_HOOK`, `NGI_SET_NODE`, `NGQF_MESG`, and `NGQF_WRITER`.

Here is the call graph for this function:



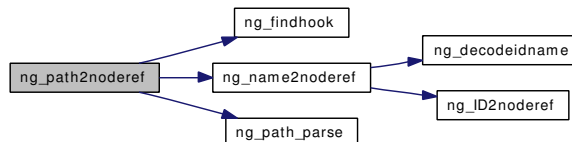
7.79.2.45 `int ng_path2noderef (node_p here, const char * address, node_p * destp, hook_p * lasthook)`

Definition at line 1589 of file ng_base.c.

References `ng_findhook()`, `NG_HOOK_NOT_VALID`, `NG_HOOK_PEER`, `ng_name2noderef()`, `NG_NODE_NOT_VALID`, `NG_NODE_REF`, `NG_NODE_UNREF`, `ng_path_parse()`, `NG_PATHSIZ`, `NG_PEER_NODE`, and `TRAP_ERROR`.

Referenced by `ng_generic_msg()`.

Here is the call graph for this function:



7.79.2.46 `int ng_path_parse (char * addr, char ** nodep, char ** pathp, char ** hookp)`

Definition at line 1520 of file ng_base.c.

Referenced by `ng_path2noderef()`.

7.79.2.47 `static __inline void ng_queue_rw (struct ng_queue * ngq, item_p item, int rw)`
`[static]`

Definition at line 1974 of file `ng_base.c`.

References `ng_queue::last`, `ng_node::nd_ID`, `NEXT_QUEUED_ITEM_CAN_PROCEED`, `ng_setisr()`, `NGI_SET_READER`, `NGI_SET_WRITER`, `NGQRW_W`, `OP_PENDING`, `ng_queue::q_flags`, `ng_queue::q_mtx`, `ng_queue::q_node`, and `ng_queue::queue`.

Referenced by `ng_acquire_read()`, `ng_acquire_write()`, and `ng_snd_item()`.

Here is the call graph for this function:



7.79.2.48 `void ng_replace_retaddr (node_p here, item_p item, ng_ID_t retaddr)`

Definition at line 3644 of file `ng_base.c`.

References `NGI_RETADDR`.

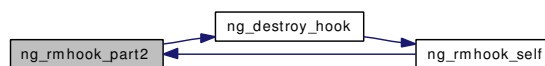
7.79.2.49 `static void ng_rmhook_part2 (node_p node, hook_p hook, void * arg1, int arg2)`
`[static]`

Definition at line 1485 of file `ng_base.c`.

References `ng_destroy_hook()`.

Referenced by `ng_rmhook_self()`.

Here is the call graph for this function:



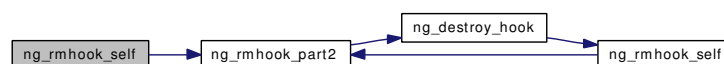
7.79.2.50 `int ng_rmhook_self (hook_p hook)`

Definition at line 1492 of file `ng_base.c`.

References `ng_deadnode`, `NG_HOOK_NODE`, `ng_rmhook_part2()`, and `ng_send_fn`.

Referenced by `ng_destroy_hook()`, `ng_pppoe_rcvdata()`, and `pppoe_ticker()`.

Here is the call graph for this function:



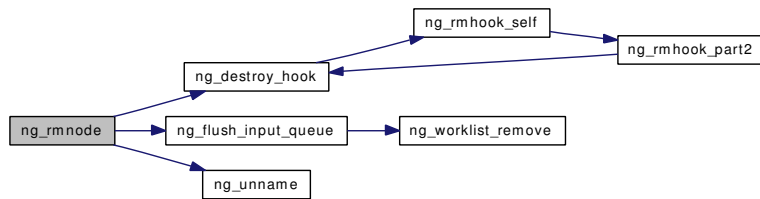
7.79.2.51 void ng_rmnode (node_p node, hook_p dummy1, void * dummy2, int dummy3)

Definition at line 663 of file ng_base.c.

References ng_type::close, ng_node::nd_flags, ng_node::nd_input_queue, ng_node::nd_type, ng_deadnode, ng_destroy_hook(), ng_flush_input_queue(), NG_NODE_IS_VALID, NG_NODE_REF, NG_NODE_UNREF, ng_unname(), NGF_CLOSING, NGF_INVALID, and ng_type::shutdown.

Referenced by ng_apply_item(), ng_generic_msg(), ng_mkpeer(), and ng_rmnode_self().

Here is the call graph for this function:



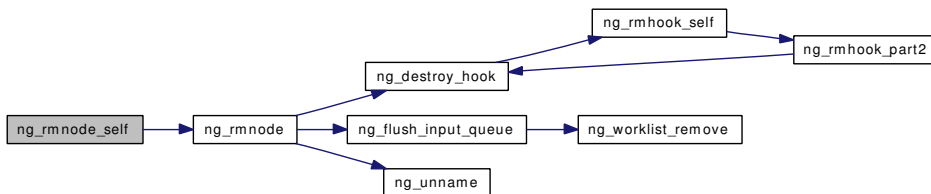
7.79.2.52 int ng_rmnode_self (node_p node)

Definition at line 1470 of file ng_base.c.

References ng_node::nd_flags, ng_deadnode, ng_rmnode(), ng_send_fn, NGF_CLOSING, and NGF_INVALID.

Referenced by bt3c_pccard_detach(), cisco_disconnect(), ng_atm_detach(), ng_atmllc_disconnect(), ng_bpf_disconnect(), ng_bridge_disconnect(), ng_ccatm_disconnect(), ng_deflate_disconnect(), ng_device_disconnect(), ng_ether_detach(), ng_ether_disconnect(), ng_gif_detach(), ng_gif_disconnect(), ng_h4_close(), ng_hci_disconnect(), ng_hub_disconnect(), ng_ksocket_disconnect(), ng_ksocket_shutdown(), ng_l2cap_disconnect(), ng_l2tp_disconnect(), ng_mppc_disconnect(), ng_nat_disconnect(), ng_netflow_disconnect(), ng_one2many_disconnect(), ng_ppp_disconnect(), ng_pppoe_disconnect(), ng_pptpgre_disconnect(), ng_pred1_disconnect(), ng_rfc1490_disconnect(), ng_socket_free_priv(), ng_source_disconnect(), ng_split_disconnect(), ng_sscfu_disconnect(), ng_sscop_disconnect(), ng_tag_disconnect(), ng_tcpmss_disconnect(), ng_UI_disconnect(), ng_uni_disconnect(), ng_vjc_disconnect(), ng_vlan_disconnect(), ng_xxx_disconnect(), nga_disconnect(), nge_disconnect(), ngfrm_disconnect(), ngh_disconnect(), ngipi_disconnect(), nglmi_disconnect(), ngs_disconnect(), ngt_close(), ngt_disconnect(), and USB_DETACH().

Here is the call graph for this function:



7.79.2.53 int ng_rmtyp (struct ng_type * tp)

Definition at line 1180 of file ng_base.c.

References `ng_type::refs`, and `TRAP_ERROR`.

Referenced by `bt3c_modevent()`, and `ubt_modevent()`.

7.79.2.54 `int ng_send_fn1 (node_p node, hook_p hook, ng_item_fn *fn, void *arg1, int arg2, int flags)`

Definition at line 3554 of file `ng_base.c`.

Referenced by `ng_ksocket_incoming()`.

7.79.2.55 `static void ng_setisr (node_p node) [static]`

Definition at line 3313 of file `ng_base.c`.

References `ng_node::nd_flags`, `ng_node::nd_ID`, `ng_node::nd_input_queue`, `NG_NODE_REF`, `NGF_WORKQ`, and `ng_queue::q_mtx`.

Referenced by `ng_dequeue()`, `ng_queue_rw()`, and `ng_snd_item()`.

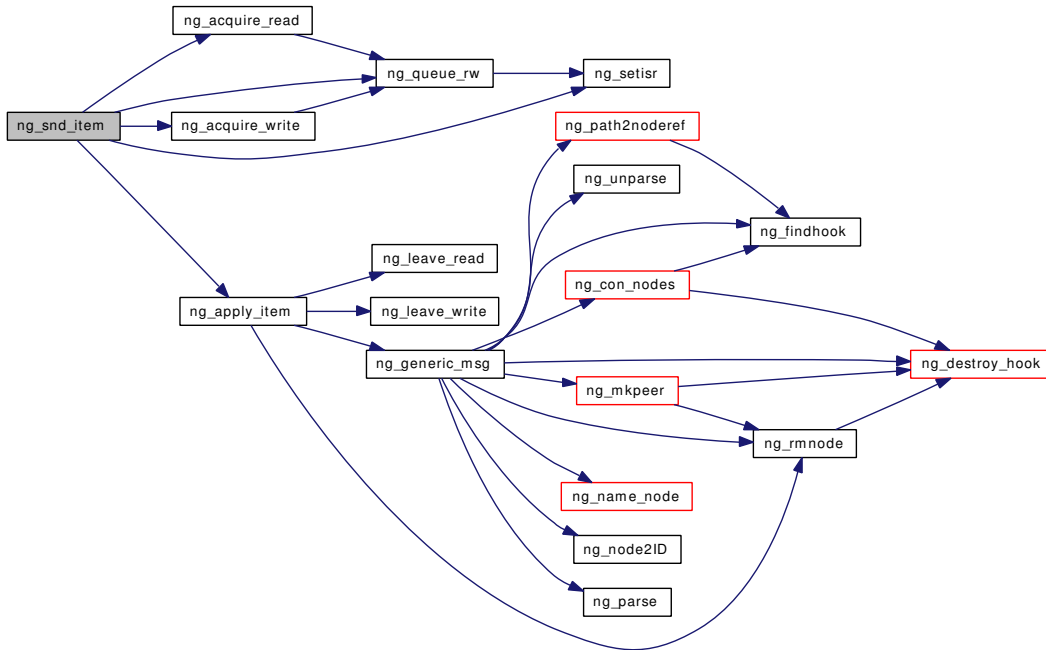
7.79.2.56 `int ng_snd_item (item_p item, int flags)`

Definition at line 2170 of file `ng_base.c`.

References `CHECK_DATA_MBUF`, `ng_item::el_flags`, `ng_hook::hk_flags`, `HK_FORCE_WRITER`, `HK_QUEUE`, `ng_node::nd_flags`, `ng_node::nd_input_queue`, `NEXT_QUEUED_ITEM_CAN_PROCEED`, `ng_acquire_read()`, `ng_acquire_write()`, `ng_apply_item()`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_HOOK_NOT_VALID`, `NG_NODE_NOT_VALID`, `NG_NODE_UNREF`, `NG_PROGRESS`, `NG_QUEUE`, `ng_queue_rw()`, `ng_setisr()`, `NGF_FORCE_WRITER`, `NGI_GET_NODE`, `NGI_HOOK`, `NGI_M`, `NGI_NODE`, `NGQF_DATA`, `NGQF_FN`, `NGQF_MESG`, `NGQF_READER`, `NGQF_RW`, `NGQF_TYPE`, `NGQF_WRITER`, `NGQRW_R`, `NGQRW_W`, `ng_queue::q_mtx`, `ng_queue::q_node`, and `TRAP_ERROR`.

Referenced by `ng_callout_trampoline()`, and `ngc_send()`.

Here is the call graph for this function:



7.79.2.57 `int ng_uncallout (struct callout * c, node_p node)`

Definition at line 3615 of file `ng_base.c`.

Referenced by `cisco_disconnect()`, `ng_bridge_shutdown()`, `ng_h4_untimeout()`, `ng_hci_command_untimeout()`, `ng_hci_con_untimeout()`, `ng_l2cap_command_untimeout()`, `ng_l2cap_discon_untimeout()`, `ng_l2cap_lp_untimeout()`, `ng_l2tp_seq_reset()`, `ng_l2tp_shutdown()`, `ng_l2tp_xmit_ctrl()`, `ng_ppp_stop_frag_timer()`, `ng_pppoe_disconnect()`, `ng_pppoe_rcvdata()`, `ng_pptpgre_stop_recv_ack_timer()`, `ng_pptpgre_stop_send_ack_timer()`, `ng_source_stop()`, `nglmi_disconnect()`, and `ngt_close()`.

7.79.2.58 `void ng_unname (node_p node)`

Definition at line 907 of file `ng_base.c`.

Referenced by `ng_rmnode()`.

7.79.2.59 `void ng_unref_hook (hook_p hook)`

Definition at line 922 of file `ng_base.c`.

References `_NG_HOOK_NODE`, `_NG_NODE_UNREF`, `ng_hook::hk_refs`, `ng_deadhook`, and `NG_FREE_HOOK`.

7.79.2.60 `int ng_unref_node (node_p node)`

Definition at line 741 of file `ng_base.c`.

References `ng_node::nd_input_queue`, `ng_node::nd_refs`, `ng_node::nd_type`, `ng_deadnode`, `NG_FREE_NODE`, `ng_queue::q_mtx`, and `ng_type::refs`.

7.79.2.61 `static void ng_worklist_remove (node_p node)` [static]

Definition at line 3290 of file ng_base.c.

References `ng_node::nd_flags`, `ng_node::nd_ID`, `ng_node::nd_input_queue`, `NG_NODE_UNREF`, `NGF_WORKQ`, and `ng_queue::q_mtx`.

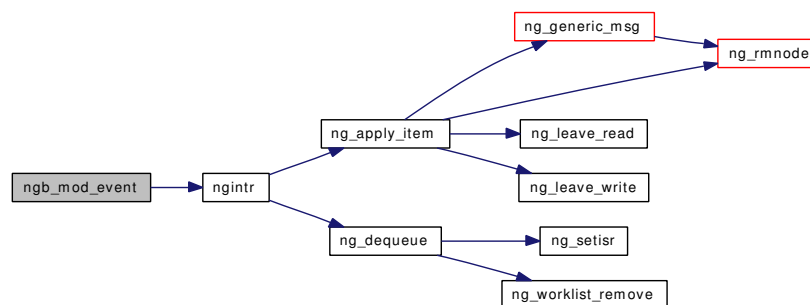
Referenced by `ng_dequeue()`, and `ng_flush_input_queue()`.

7.79.2.62 `static int ngb_mod_event (module_t mod, int event, void * data)` [static]

Definition at line 3061 of file ng_base.c.

References `ngintr()`.

Here is the call graph for this function:

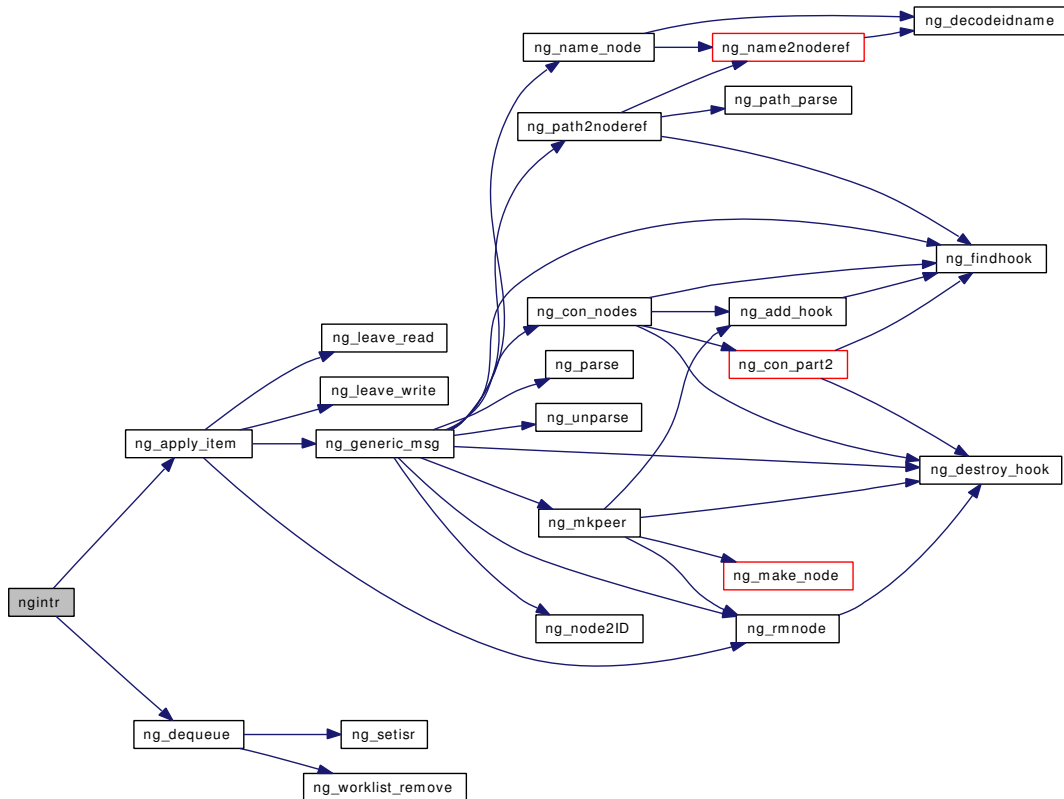
**7.79.2.63** `static void ngintr (void)` [static]

Definition at line 3240 of file ng_base.c.

References `ng_node::nd_flags`, `ng_node::nd_ID`, `ng_node::nd_input_queue`, `ng_apply_item()`, `ng_dequeue()`, `NG_NODE_UNREF`, `NGF_WORKQ`, `NGI_GET_NODE`, and `ng_queue::q_mtx`.

Referenced by `ngb_mod_event()`.

Here is the call graph for this function:



7.79.2.64 `SYSCALL_INT` (`_net_graph`, `OID_AUTO`, `msg_version`, `CTLFLAG_RD`, `0`, `NG_VERSION`, `""`)

7.79.2.65 `SYSCALL_INT` (`_net_graph`, `OID_AUTO`, `abi_version`, `CTLFLAG_RD`, `0`, `NG_ABI_VERSION`, `""`)

7.79.2.66 `SYSCALL_INT` (`_net_graph`, `OID_AUTO`, `maxalloc`, `CTLFLAG_RDTUN`, `&maxalloc`, `0`, "Maximum number of queue items to allocate")

7.79.2.67 `SYSCALL_NODE` (`_net`, `OID_AUTO`, `graph`, `CTLFLAG_RW`, `0`, "netgraph Family")

7.79.2.68 `static TAILQ_HEAD` (`ng_node`) [`static`]

Definition at line 161 of file `ng_base.c`.

7.79.2.69 `TUNABLE_INT` ("net.graph.maxalloc", `&maxalloc`)

7.79.3 Variable Documentation

7.79.3.1 `int maxalloc = 512` [`static`]

Definition at line 2903 of file `ng_base.c`.

7.79.3.2 moduledata_t netgraph_mod [static]**Initial value:**

```
{
    "netgraph",
    ngb_mod_event,
    (NULL)
}
```

Definition at line 3098 of file ng_base.c.

7.79.3.3 struct ng_hook ng_deadhook**Initial value:**

```
{
    "dead",
    NULL,
    HK_INVALID | HK_DEAD,
    1,
    0,
    &ng_deadhook,
    &ng_deadnode,
    {},
    NULL,
    NULL,
}
```

Definition at line 138 of file ng_base.c.

Referenced by ng_add_hook(), ng_bypass(), ng_con_part2(), ng_destroy_hook(), and ng_unref_hook().

7.79.3.4 struct ng_node ng_deadnode**Initial value:**

```
{
    "dead",
    &ng_deadtype,
    NGF_INVALID,
    1,
    0,
    NULL,
    0,
    LIST_HEAD_INITIALIZER(ng_deadnode.hooks),
    {},
    {},
    {},
    {
        0,
        {},
        NULL,
        &ng_deadnode.nd_input_queue.queue,
        &ng_deadnode
    }
}
```

```

    },

}

```

Definition at line 112 of file ng_base.c.

Referenced by ng_acquire_read(), ng_acquire_write(), ng_con_nodes(), ng_con_part3(), ng_destroy_hook(), ng_rmhook_self(), ng_rmnnode(), ng_rmnnode_self(), and ng_unref_node().

7.79.3.5 struct `ng_cmdlist ng_generic_cmds[]` [static]

Definition at line 431 of file ng_base.c.

Referenced by ng_generic_msg().

7.79.3.6 struct `ng_parse_type ng_generic_linkinfo_array_type` [static]

Initial value:

```

{
    &ng_parse_array_type,
    &ng_generic_linkinfo_array_type_info
}

```

Definition at line 419 of file ng_base.c.

7.79.3.7 struct `ng_parse_array_info ng_generic_linkinfo_array_type_info` [static]

Initial value:

```

{
    &ng_generic_linkinfo_type,
    &ng_generic_linkinfo_getLength
}

```

Definition at line 415 of file ng_base.c.

7.79.3.8 struct `ng_parse_type ng_generic_nodeinfoarray_type` [static]

Initial value:

```

{
    &ng_parse_array_type,
    &ng_nodeinfoarray_type_info
}

```

Definition at line 399 of file ng_base.c.

7.79.3.9 struct [ng_parse_type](#) [ng_generic_typeinfoarray_type](#) [static]**Initial value:**

```
{
    &ng_parse_array_type,
    &ng_typeinfoarray_type_info
}
```

Definition at line 409 of file ng_base.c.

7.79.3.10 struct [ng_parse_array_info](#) [ng_nodeinfoarray_type_info](#) [static]**Initial value:**

```
{
    &ng_generic_nodeinfo_type,
    &ng_generic_list_getLength
}
```

Definition at line 395 of file ng_base.c.

7.79.3.11 [uma_zone_t](#) [ng_qzone](#)

Definition at line 2902 of file ng_base.c.

7.79.3.12 struct [ng_parse_array_info](#) [ng_typeinfoarray_type_info](#) [static]**Initial value:**

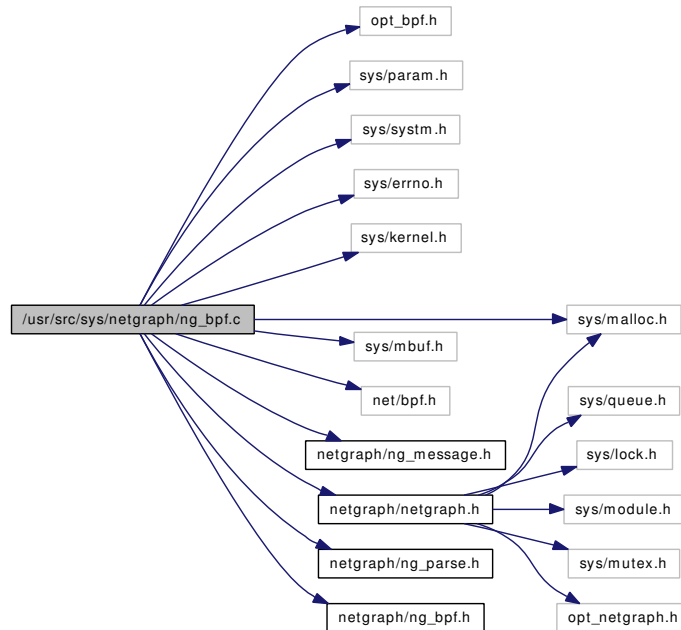
```
{
    &ng_generic_typeinfo_type,
    &ng_generic_list_getLength
}
```

Definition at line 405 of file ng_base.c.

7.80 /usr/src/sys/netgraph/ng_bpf.c File Reference

```
#include "opt_bpf.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <net/bpf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_bpf.h>
```

Include dependency graph for ng_bpf.c:



Data Structures

- struct [ng_bpf_hookinfo](#)

Defines

- #define [M_NETGRAPH_BPF](#) M_NETGRAPH
- #define [OFFSETOF](#)(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))

- #define `ERROUT(x)` do { error = (x); goto done; } while (0)

Typedefs

- typedef `ng_bpf_hookinfo * hinfo_p`

Functions

- static int `ng_bpf_setprog` (`hook_p` hook, const struct `ng_bpf_hookprog *hp`)
- static int `ng_bpf_hookprogary_getLength` (const struct `ng_parse_type *type`, const `u_char *start`, const `u_char *buf`)
- `NETGRAPH_INIT` (`bpf,&typestruct`)
- static int `ng_bpf_constructor` (`node_p` node)
- static int `ng_bpf_newhook` (`node_p` node, `hook_p` hook, const char *`name`)
- static int `ng_bpf_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_bpf_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_bpf_shutdown` (`node_p` node)
- static int `ng_bpf_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_bpf_constructor`
- static `ng_rcvmsg_t` `ng_bpf_rcvmsg`
- static `ng_shutdown_t` `ng_bpf_shutdown`
- static `ng_newhook_t` `ng_bpf_newhook`
- static `ng_rcvdata_t` `ng_bpf_rcvdata`
- static `ng_disconnect_t` `ng_bpf_disconnect`
- static struct `ng_parse_struct_field` `ng_bpf_insn_type_fields` []
- static struct `ng_parse_type` `ng_bpf_insn_type`
- static struct `ng_parse_array_info` `ng_bpf_hookprogary_info`
- static struct `ng_parse_type` `ng_bpf_hookprogary_type`
- static struct `ng_parse_struct_field` `ng_bpf_hookprog_type_fields` [] = `NG_BPF_HOOKPROG_-TYPE_INFO(&ng_bpf_hookprogary_type)`
- static struct `ng_parse_type` `ng_bpf_hookprog_type`
- static struct `ng_parse_struct_field` `ng_bpf_hookstat_type_fields` [] = `NG_BPF_HOOKSTAT_-TYPE_INFO`
- static struct `ng_parse_type` `ng_bpf_hookstat_type`
- static struct `ng_cmdlist` `ng_bpf_cmdlist` []
- static struct `ng_type` `typestruct`
- static struct `ng_bpf_hookprog` `ng_bpf_default_prog`

7.80.1 Define Documentation

7.80.1.1 #define `ERROUT(x)` do { error = (x); goto done; } while (0)

Definition at line 84 of file `ng_bpf.c`.

7.80.1.2 #define M_NETGRAPH_BPF M_NETGRAPH

Definition at line 79 of file ng_bpf.c.

Referenced by ng_bpf_disconnect(), ng_bpf_newhook(), ng_bpf_rcvdata(), and ng_bpf_setprog().

7.80.1.3 #define OFFSETOF(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))

Definition at line 82 of file ng_bpf.c.

Referenced by ng_bpf_hookprogary_getLength(), ng_ksocket_finish_accept(), ng_ksocket_sockaddr_parse(), ng_ksocket_sockaddr_unparse(), and ng_parse_sockoptval_getLength().

7.80.2 Typedef Documentation

7.80.2.1 typedef struct ng_bpf_hookinfo* hinfo_p

Definition at line 96 of file ng_bpf.c.

7.80.3 Function Documentation

7.80.3.1 NETGRAPH_INIT (bpf, & typestruct)

7.80.3.2 static int ng_bpf_constructor (node_p node) [static]

Definition at line 230 of file ng_bpf.c.

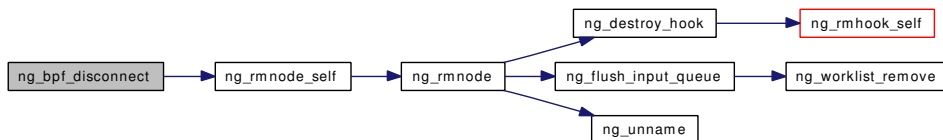
References NG_NODE_SET_PRIVATE.

7.80.3.3 static int ng_bpf_disconnect (hook_p hook) [static]

Definition at line 472 of file ng_bpf.c.

References M_NETGRAPH_BPF, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, ng_rmnode_self(), and ng_bpf_hookinfo::prog.

Here is the call graph for this function:



7.80.3.4 static int ng_bpf_hookprogary_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]

Definition at line 124 of file ng_bpf.c.

References ng_bpf_hookprog::bpf_prog, ng_bpf_hookprog::bpf_prog_len, and OFFSETOF.

7.80.3.5 `static int ng_bpf_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 240 of file ng_bpf.c.

References `M_NETGRAPH_BPF`, `ng_bpf_default_prog`, `ng_bpf_setprog()`, and `NG_HOOK_SET_PRIVATE`.

Here is the call graph for this function:

**7.80.3.6** `static int ng_bpf_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 380 of file ng_bpf.c.

References `ng_bpf_hookprog::bpf_prog`, `ng_bpf_hookprog::ifMatch`, `ng_bpf_hookprog::ifNotMatch`, `M_NETGRAPH_BPF`, `ng_findhook()`, `NG_FREE_ITEM`, `NG_FWD_ITEM_HOOK`, `NG_HOOK_PRIVATE`, `NGI_M`, `ng_bpf_hookinfo::node`, `ng_bpf_hookinfo::prog`, `ng_bpf_hookstat::recvFrames`, `ng_bpf_hookstat::recvMatchFrames`, `ng_bpf_hookstat::recvMatchOctets`, `ng_bpf_hookstat::recvOctets`, `ng_bpf_hookinfo::stats`, `ng_bpf_hookstat::xmitFrames`, and `ng_bpf_hookstat::xmitOctets`.

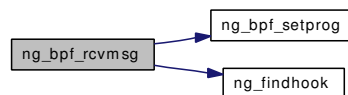
Here is the call graph for this function:

**7.80.3.7** `static int ng_bpf_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 270 of file ng_bpf.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_bpf_hookprog::bpf_prog_len`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ERROUT`, `ng_mesg::header`, `NG_BPF_HOOKPROG_SIZE`, `ng_bpf_setprog()`, `ng_findhook()`, `NG_FREE_MSG`, `NG_HOOK_PRIVATE`, `NG_MKRESPONSE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_BPF_CLR_STATS`, `NGM_BPF_COOKIE`, `NGM_BPF_GET_PROGRAM`, `NGM_BPF_GET_STATS`, `NGM_BPF_GETCLR_STATS`, `NGM_BPF_SET_PROGRAM`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:

**7.80.3.8** `static int ng_bpf_setprog (hook_p hook, const struct ng_bpf_hookprog * hp)` [static]

Definition at line 500 of file ng_bpf.c.

References `ng_bpf_hookprog::bpf_prog`, `ng_bpf_hookprog::bpf_prog_len`, `M_NETGRAPH_BPF`, `NG_BPF_HOOKPROG_SIZE`, `NG_HOOK_PRIVATE`, and `ng_bpf_hookinfo::prog`.

Referenced by `ng_bpf_newhook()`, and `ng_bpf_rcvmsg()`.

7.80.3.9 `static int ng_bpf_shutdown (node_p node)` [static]

Definition at line 462 of file ng_bpf.c.

References NG_NODE_UNREF.

7.80.4 Variable Documentation**7.80.4.1** `struct ng_cmdlist ng_bpf_cmdlist[]` [static]

Definition at line 161 of file ng_bpf.c.

7.80.4.2 `ng_constructor_t ng_bpf_constructor` [static]

Definition at line 99 of file ng_bpf.c.

7.80.4.3 `struct ng_bpf_hookprog ng_bpf_default_prog` [static]**Initial value:**

```
{
    { '\0' },
    { '\0' },
    { '\0' },
    1,
    { BPF_STMT (BPF_RET+BPF_K, 0) }
}
```

Definition at line 215 of file ng_bpf.c.

Referenced by ng_bpf_newhook().

7.80.4.4 `ng_disconnect_t ng_bpf_disconnect` [static]

Definition at line 104 of file ng_bpf.c.

7.80.4.5 `struct ng_parse_type ng_bpf_hookprog_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bpf_hookprog_type_fields
}
```

Definition at line 147 of file ng_bpf.c.

7.80.4.6 `struct ng_parse_struct_field ng_bpf_hookprog_type_fields[] =
NG_BPF_HOOKPROG_TYPE_INFO(&ng_bpf_hookprog_type)` [static]

Definition at line 146 of file ng_bpf.c.

7.80.4.7 struct [ng_parse_array_info](#) [ng_bpf_hookprogary_info](#) [static]**Initial value:**

```
{
    &ng_bpf_insn_type,
    &ng_bpf_hookprogary_getLength,
    NULL
}
```

Definition at line 134 of file ng_bpf.c.

7.80.4.8 struct [ng_parse_type](#) [ng_bpf_hookprogary_type](#) [static]**Initial value:**

```
{
    &ng_parse_array_type,
    &ng_bpf_hookprogary_info
}
```

Definition at line 139 of file ng_bpf.c.

7.80.4.9 struct [ng_parse_type](#) [ng_bpf_hookstat_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bpf_hookstat_type_fields
}
```

Definition at line 155 of file ng_bpf.c.

7.80.4.10 struct [ng_parse_struct_field](#) [ng_bpf_hookstat_type_fields](#)[] =
[NG_BPF_HOOKSTAT_TYPE_INFO](#) [static]

Definition at line 154 of file ng_bpf.c.

7.80.4.11 struct [ng_parse_type](#) [ng_bpf_insn_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bpf_insn_type_fields
}
```

Definition at line 117 of file ng_bpf.c.

7.80.4.12 `struct ng_parse_struct_field ng_bpf_insn_type_fields[]` [static]**Initial value:**

```
{
    { "code",      &ng_parse_hint16_type  },
    { "jt",       &ng_parse_uint8_type   },
    { "jf",       &ng_parse_uint8_type   },
    { "k",        &ng_parse_uint32_type  },
    { NULL }
}
```

Definition at line 110 of file ng_bpf.c.

7.80.4.13 `ng_newhook_t ng_bpf_newhook` [static]

Definition at line 102 of file ng_bpf.c.

7.80.4.14 `ng_rcvdata_t ng_bpf_rcvdata` [static]

Definition at line 103 of file ng_bpf.c.

7.80.4.15 `ng_rcvmsg_t ng_bpf_rcvmsg` [static]

Definition at line 100 of file ng_bpf.c.

7.80.4.16 `ng_shutdown_t ng_bpf_shutdown` [static]

Definition at line 101 of file ng_bpf.c.

7.80.4.17 `struct ng_type_tpestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_BPF_NODE_TYPE,
    .constructor = ng_bpf_constructor,
    .rcvmsg =      ng_bpf_rcvmsg,
    .shutdown =    ng_bpf_shutdown,
    .newhook =     ng_bpf_newhook,
    .rcvdata =     ng_bpf_rcvdata,
    .disconnect =  ng_bpf_disconnect,
    .cmdlist =     ng_bpf_cmdlist,
}
```

Definition at line 201 of file ng_bpf.c.

7.81 /usr/src/sys/netgraph/ng_bpf.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_bpf_hookprog](#)
- struct [ng_bpf_hookstat](#)

Defines

- #define [NG_BPF_NODE_TYPE](#) "bpf"
- #define [NGM_BPF_COOKIE](#) 944100792
- #define [NG_BPF_HOOKPROG_SIZE](#)(numInsn) (sizeof(struct [ng_bpf_hookprog](#)) + (numInsn) * sizeof(struct [bpf_insn](#)))
- #define [NG_BPF_HOOKPROG_TYPE_INFO](#)(bptype)
- #define [NG_BPF_HOOKSTAT_TYPE_INFO](#)

Enumerations

- enum {
 - [NGM_BPF_SET_PROGRAM](#) = 1, [NGM_BPF_GET_PROGRAM](#), [NGM_BPF_GET_STATS](#),
 - [NGM_BPF_CLR_STATS](#),
 - [NGM_BPF_GETCLR_STATS](#) }

7.81.1 Define Documentation

7.81.1.1 #define [NG_BPF_HOOKPROG_SIZE](#)(numInsn) (sizeof(struct [ng_bpf_hookprog](#)) + (numInsn) * sizeof(struct [bpf_insn](#)))

Definition at line 60 of file [ng_bpf.h](#).

Referenced by [ng_bpf_rcvmsg\(\)](#), and [ng_bpf_setprog\(\)](#).

7.81.1.2 #define [NG_BPF_HOOKPROG_TYPE_INFO](#)(bptype)

Value:

```

{
    \
    { "thisHook",          &ng_parse_hookbuf_type }, \
    { "ifMatch",          &ng_parse_hookbuf_type }, \
    { "ifNotMatch",      &ng_parse_hookbuf_type }, \
    { "bpf_prog_len",     &ng_parse_int32_type  }, \
    { "bpf_prog",        (bptype)                }, \
    { NULL }
}
  
```

Definition at line 64 of file [ng_bpf.h](#).

7.81.1.3 #define NG_BPF_HOOKSTAT_TYPE_INFO**Value:**

```

{
    { "recvFrames",      &ng_parse_uint64_type }, \
    { "recvOctets",     &ng_parse_uint64_type }, \
    { "recvMatchFrames", &ng_parse_uint64_type }, \
    { "recvMatchOctets", &ng_parse_uint64_type }, \
    { "xmitFrames",     &ng_parse_uint64_type }, \
    { "xmitOctets",     &ng_parse_uint64_type }, \
    { NULL }
}

```

Definition at line 84 of file ng_bpf.h.

7.81.1.4 #define NG_BPF_NODE_TYPE "bpf"

Definition at line 48 of file ng_bpf.h.

7.81.1.5 #define NGM_BPF_COOKIE 944100792

Definition at line 49 of file ng_bpf.h.

Referenced by ng_bpf_rcvmsg().

7.81.2 Enumeration Type Documentation**7.81.2.1 anonymous enum****Enumerator:**

```

    NGM_BPF_SET_PROGRAM
    NGM_BPF_GET_PROGRAM
    NGM_BPF_GET_STATS
    NGM_BPF_CLR_STATS
    NGM_BPF_GETCLR_STATS

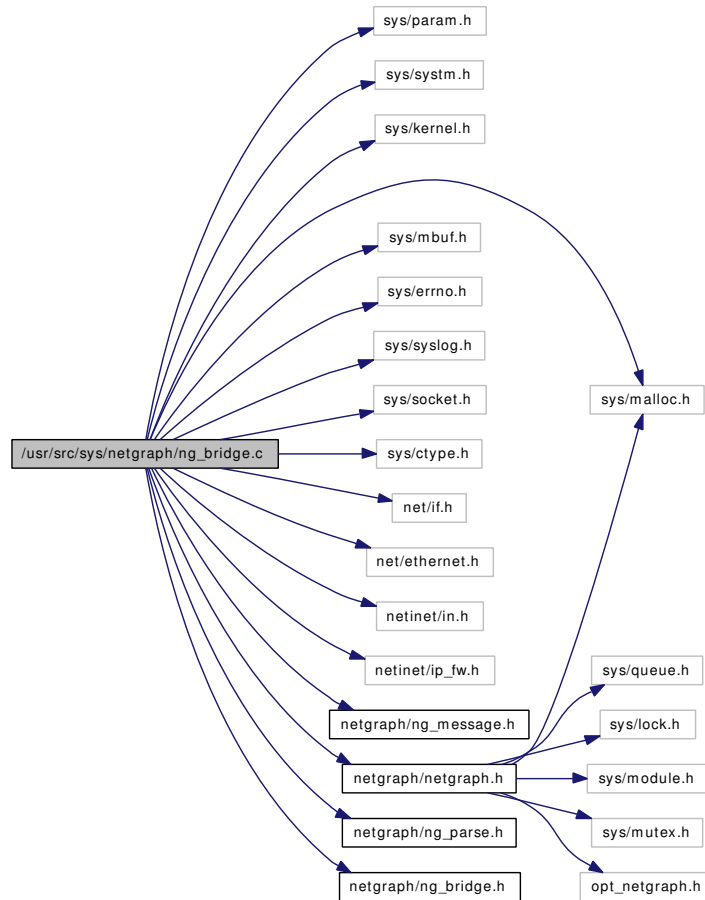
```

Definition at line 95 of file ng_bpf.h.

7.82 /usr/src/sys/netgraph/ng_bridge.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <sys/ctype.h>
#include <net/if.h>
#include <net/ethernet.h>
#include <netinet/in.h>
#include <netinet/ip_fw.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_bridge.h>
```

Include dependency graph for ng_bridge.c:



Data Structures

- struct [ng_bridge_link](#)
- struct [ng_bridge_private](#)
- struct [ng_bridge_hent](#)

Defines

- #define [M_NETGRAPH_BRIDGE](#) M_NETGRAPH
- #define [LINK_NUM](#)(hook) (*(u_int16_t *)&(hook) → private)
- #define [ETHER_EQUAL](#)(a, b)
- #define [MIN_BUCKETS](#) (1 << 5)
- #define [MAX_BUCKETS](#) (1 << 14)
- #define [DEFAULT_LOOP_TIMEOUT](#) 60
- #define [DEFAULT_MAX_STALENESS](#) (15 * 60)
- #define [DEFAULT_MIN_STABLE_AGE](#) 1
- #define [HASH](#)(addr, mask)

Typedefs

- typedef [ng_bridge_private](#) * [priv_p](#)

Functions

- [SLIST_HEAD](#) ([ng_bridge_bucket](#), [ng_bridge_hent](#))
- static struct [ng_bridge_host](#) * [ng_bridge_get](#) ([priv_p](#) [priv](#), const u_char *[addr](#))
- static int [ng_bridge_put](#) ([priv_p](#) [priv](#), const u_char *[addr](#), int [linkNum](#))
- static void [ng_bridge_rehash](#) ([priv_p](#) [priv](#))
- static void [ng_bridge_remove_hosts](#) ([priv_p](#) [priv](#), int [linkNum](#))
- static void [ng_bridge_timeout](#) ([node_p](#) [node](#), [hook_p](#) [hook](#), void *[arg1](#), int [arg2](#))
- static const char * [ng_bridge_nodename](#) ([node_p](#) [node](#))
- static int [ng_bridge_getTableLength](#) (const struct [ng_parse_type](#) *[type](#), const u_char *[start](#), const u_char *[buf](#))
- [NETGRAPH_INIT](#) ([bridge](#), &[ng_bridge_tpestruct](#))
- static int [ng_bridge_constructor](#) ([node_p](#) [node](#))
- static int [ng_bridge_newhook](#) ([node_p](#) [node](#), [hook_p](#) [hook](#), const char *[name](#))
- static int [ng_bridge_rcvmsg](#) ([node_p](#) [node](#), [item_p](#) [item](#), [hook_p](#) [lasthook](#))
- static int [ng_bridge_rcvdata](#) ([hook_p](#) [hook](#), [item_p](#) [item](#))
- static int [ng_bridge_shutdown](#) ([node_p](#) [node](#))
- static int [ng_bridge_disconnect](#) ([hook_p](#) [hook](#))

Variables

- static [ng_constructor_t](#) [ng_bridge_constructor](#)
- static [ng_rcvmsg_t](#) [ng_bridge_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_bridge_shutdown](#)
- static [ng_newhook_t](#) [ng_bridge_newhook](#)
- static [ng_rcvdata_t](#) [ng_bridge_rcvdata](#)
- static [ng_disconnect_t](#) [ng_bridge_disconnect](#)
- static const u_char [ng_bridge_bcast_addr](#) [[ETHER_ADDR_LEN](#)]
- static struct [ng_parse_struct_field](#) [ng_bridge_host_type_fields](#) [] = [NG_BRIDGE_HOST_TYPE_INFO](#)(&[ng_parse_enaddr_type](#))
- static struct [ng_parse_type](#) [ng_bridge_host_type](#)
- static struct [ng_parse_array_info](#) [ng_bridge_hary_type_info](#)
- static struct [ng_parse_type](#) [ng_bridge_hary_type](#)
- static struct [ng_parse_struct_field](#) [ng_bridge_host_ary_type_fields](#) [] = [NG_BRIDGE_HOST_ARY_TYPE_INFO](#)(&[ng_bridge_hary_type](#))
- static struct [ng_parse_type](#) [ng_bridge_host_ary_type](#)
- static struct [ng_parse_fixedarray_info](#) [ng_bridge_ipfwary_type_info](#)
- static struct [ng_parse_type](#) [ng_bridge_ipfwary_type](#)
- static struct [ng_parse_struct_field](#) [ng_bridge_config_type_fields](#) [] = [NG_BRIDGE_CONFIG_TYPE_INFO](#)(&[ng_bridge_ipfwary_type](#))
- static struct [ng_parse_type](#) [ng_bridge_config_type](#)
- static struct [ng_parse_struct_field](#) [ng_bridge_stats_type_fields](#) [] = [NG_BRIDGE_STATS_TYPE_INFO](#)
- static struct [ng_parse_type](#) [ng_bridge_stats_type](#)
- static struct [ng_cmdlist](#) [ng_bridge_cmdlist](#) []
- static struct [ng_type](#) [ng_bridge_tpestruct](#)

7.82.1 Define Documentation

7.82.1.1 #define DEFAULT_LOOP_TIMEOUT 60

Definition at line 152 of file ng_bridge.c.

Referenced by ng_bridge_constructor().

7.82.1.2 #define DEFAULT_MAX_STALENESS (15 * 60)

Definition at line 153 of file ng_bridge.c.

Referenced by ng_bridge_constructor().

7.82.1.3 #define DEFAULT_MIN_STABLE_AGE 1

Definition at line 154 of file ng_bridge.c.

Referenced by ng_bridge_constructor().

7.82.1.4 #define ETHER_EQUAL(a, b)

Value:

```
((const u_int32_t *) (a)) [0] \
                                     == ((const u_int32_t *) (b)) [0] \
&& ((const u_int16_t *) (a)) [2] \
                                     == ((const u_int16_t *) (b)) [2])
```

Definition at line 142 of file ng_bridge.c.

Referenced by ng_bridge_get(), ng_bridge_put(), and ng_bridge_rcvdata().

7.82.1.5 #define HASH(addr, mask)

Value:

```
( (((const u_int16_t *) (addr)) [0] \
                                     \
                                     ^ ((const u_int16_t *) (addr)) [1] \
                                     \
                                     ^ ((const u_int16_t *) (addr)) [2]) & (mask) )
```

Definition at line 812 of file ng_bridge.c.

Referenced by ng_bridge_get(), ng_bridge_put(), ng_bridge_rehash(), ng_etf_findentry(), ng_etf_rcvmsg(), ng_vlan_findentry(), ng_vlan_rcvmsg(), Pred1Compress(), Pred1Decompress(), and Pred1SyncTable().

7.82.1.6 #define LINK_NUM(hook) (*(u_int16_t *)&(hook) → private)

Definition at line 139 of file ng_bridge.c.

7.82.1.7 #define M_NETGRAPH_BRIDGE M_NETGRAPH

Definition at line 85 of file ng_bridge.c.

Referenced by ng_bridge_constructor(), ng_bridge_disconnect(), ng_bridge_newhook(), ng_bridge_put(), ng_bridge_rehash(), ng_bridge_remove_hosts(), ng_bridge_shutdown(), and ng_bridge_timeout().

7.82.1.8 #define MAX_BUCKETS (1 << 14)

Definition at line 149 of file ng_bridge.c.

Referenced by ng_bridge_rehash().

7.82.1.9 #define MIN_BUCKETS (1 << 5)

Definition at line 148 of file ng_bridge.c.

Referenced by ng_bridge_constructor(), and ng_bridge_rehash().

7.82.2 Typedef Documentation**7.82.2.1 typedef struct ng_bridge_private* priv_p**

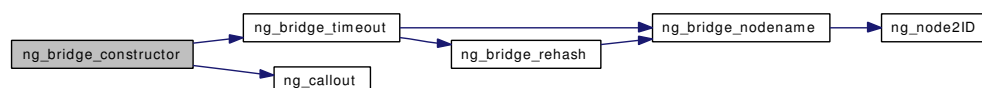
Definition at line 107 of file ng_bridge.c.

7.82.3 Function Documentation**7.82.3.1 NETGRAPH_INIT (bridge, & ng_bridge_tpestruct)****7.82.3.2 static int ng_bridge_constructor (node_p node) [static]**

Definition at line 295 of file ng_bridge.c.

References DEFAULT_LOOP_TIMEOUT, DEFAULT_MAX_STALENESS, DEFAULT_MIN_STABLE_AGE, M_NETGRAPH_BRIDGE, MIN_BUCKETS, ng_bridge_timeout(), ng_callout(), ng_callout_init, NG_NODE_FORCE_WRITER, and NG_NODE_SET_PRIVATE.

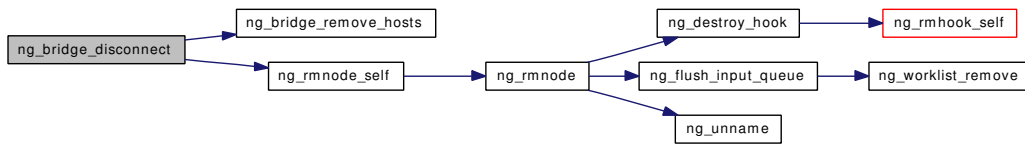
Here is the call graph for this function:

**7.82.3.3 static int ng_bridge_disconnect (hook_p hook) [static]**

Definition at line 778 of file ng_bridge.c.

References M_NETGRAPH_BRIDGE, NG_BRIDGE_MAX_LINKS, ng_bridge_remove_hosts(), NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.82.3.4 static struct **ng_bridge_host** * **ng_bridge_get** (**priv_p** *priv*, const u_char * *addr*) [static]

Definition at line 820 of file ng_bridge.c.

References ng_bridge_host::addr, ETHER_EQUAL, HASH, and ng_bridge_hent::host.

Referenced by ng_bridge_rcvdata().

7.82.3.5 static int **ng_bridge_getTableLength** (const struct **ng_parse_type** * *type*, const u_char * *start*, const u_char * *buf*) [static]

Definition at line 164 of file ng_bridge.c.

References ng_bridge_host_ary::numHosts.

7.82.3.6 static int **ng_bridge_newhook** (**node_p** *node*, **hook_p** *hook*, const char * *name*) [static]

Definition at line 342 of file ng_bridge.c.

References M_NETGRAPH_BRIDGE, NG_BRIDGE_HOOK_LINK_PREFIX, NG_BRIDGE_MAX_LINKS, NG_HOOK_SET_PRIVATE, and NG_NODE_PRIVATE.

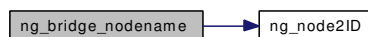
7.82.3.7 static const char * **ng_bridge_nodename** (**node_p** *node*) [static]

Definition at line 1032 of file ng_bridge.c.

References name, ng_node2ID(), NG_NODE_NAME, and NG_NODESIZ.

Referenced by ng_bridge_rcvdata(), ng_bridge_rehash(), and ng_bridge_timeout().

Here is the call graph for this function:



7.82.3.8 static int **ng_bridge_put** (**priv_p** *priv*, const u_char * *addr*, int *linkNum*) [static]

Definition at line 838 of file ng_bridge.c.

References ng_bridge_host::addr, ETHER_EQUAL, HASH, ng_bridge_hent::host, M_NETGRAPH_BRIDGE, and ng_bridge_rehash().

Referenced by `ng_bridge_rcvdata()`.

Here is the call graph for this function:

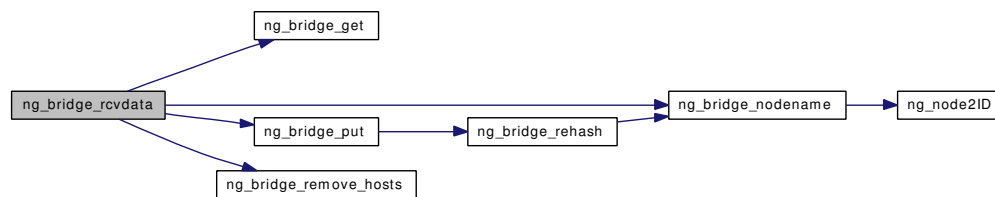


7.82.3.9 `static int ng_bridge_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 514 of file `ng_bridge.c`.

References `ng_bridge_host::age`, `ETHER_EQUAL`, `ng_bridge_link::hook`, `ng_bridge_host::linkNum`, `ng_bridge_link::loopCount`, `ng_bridge_link_stats::loopDetects`, `ng_bridge_link_stats::loopDrops`, `ng_bridge_link_stats::memoryFailures`, `ng_bridge_bcast_addr`, `ng_bridge_get()`, `NG_BRIDGE_MAX_LINKS`, `ng_bridge_nodename()`, `ng_bridge_put()`, `ng_bridge_remove_hosts()`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_HOOK_NAME`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `NGI_GET_M`, `ng_bridge_link_stats::recvBroadcasts`, `ng_bridge_link_stats::recvInvalid`, `ng_bridge_link_stats::recvMulticasts`, `ng_bridge_link_stats::recvOctets`, `ng_bridge_link_stats::recvPackets`, `ng_bridge_link_stats::recvRunts`, `ng_bridge_link_stats::recvUnknown`, `ng_bridge_host::staleness`, `ng_bridge_link::stats`, `ng_bridge_link_stats::xmitOctets`, and `ng_bridge_link_stats::xmitPackets`.

Here is the call graph for this function:



7.82.3.10 `static int ng_bridge_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 379 of file `ng_bridge.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `ng_bridge_hent::host`, `NG_BRIDGE_MAX_LINKS`, `ng_bridge_remove_hosts()`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NGI_GET_MSG`, `NGM_BRIDGE_CLR_STATS`, `NGM_BRIDGE_COOKIE`, `NGM_BRIDGE_GET_CONFIG`, `NGM_BRIDGE_GET_STATS`, `NGM_BRIDGE_GET_TABLE`, `NGM_BRIDGE_GETCLR_STATS`, `NGM_BRIDGE_RESET`, `NGM_BRIDGE_SET_CONFIG`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



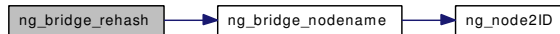
7.82.3.11 `static void ng_bridge_rehash (priv_p priv)` [static]

Definition at line 878 of file `ng_bridge.c`.

References HASH, M_NETGRAPH_BRIDGE, MAX_BUCKETS, MIN_BUCKETS, and ng_bridge_nodename().

Referenced by ng_bridge_put(), and ng_bridge_timeout().

Here is the call graph for this function:



7.82.3.12 static void ng_bridge_remove_hosts (**priv_p** *priv*, **int** *linkNum*) [static]

Definition at line 938 of file ng_bridge.c.

References M_NETGRAPH_BRIDGE.

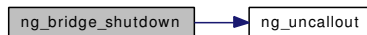
Referenced by ng_bridge_disconnect(), ng_bridge_rcvdata(), and ng_bridge_rcvmmsg().

7.82.3.13 static int ng_bridge_shutdown (**node_p** *node*) [static]

Definition at line 752 of file ng_bridge.c.

References M_NETGRAPH_BRIDGE, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_uncallout().

Here is the call graph for this function:



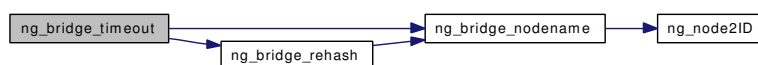
7.82.3.14 static void ng_bridge_timeout (**node_p** *node*, **hook_p** *hook*, **void *** *arg1*, **int** *arg2*) [static]

Definition at line 965 of file ng_bridge.c.

References M_NETGRAPH_BRIDGE, NG_BRIDGE_MAX_LINKS, ng_bridge_nodename(), ng_bridge_rehash(), and NG_NODE_PRIVATE.

Referenced by ng_bridge_constructor().

Here is the call graph for this function:



7.82.3.15 SLIST_HEAD (ng_bridge_bucket, ng_bridge_hent)

7.82.4 Variable Documentation

7.82.4.1 const u_char ng_bridge_bcast_addr[ETHER_ADDR_LEN] [static]

Initial value:

```
{ 0xff, 0xff, 0xff, 0xff, 0xff, 0xff }
```

Definition at line 135 of file ng_bridge.c.

Referenced by ng_bridge_rcvdata().

7.82.4.2 struct ng_cmdlist ng_bridge_cmdlist[] [static]

Definition at line 220 of file ng_bridge.c.

7.82.4.3 struct ng_parse_type ng_bridge_config_type [static]

Initial value:

```
{  
    &ng_parse_struct_type,  
    &ng_bridge_config_type_fields  
}
```

Definition at line 206 of file ng_bridge.c.

7.82.4.4 struct ng_parse_struct_field ng_bridge_config_type_fields[] = NG_BRIDGE_CONFIG_TYPE_INFO(&ng_bridge_ipfwary_type) [static]

Definition at line 205 of file ng_bridge.c.

7.82.4.5 ng_constructor_t ng_bridge_constructor [static]

Definition at line 119 of file ng_bridge.c.

7.82.4.6 ng_disconnect_t ng_bridge_disconnect [static]

Definition at line 124 of file ng_bridge.c.

7.82.4.7 struct ng_parse_type ng_bridge_hary_type [static]

Initial value:

```
{  
    &ng_parse_array_type,  
    &ng_bridge_hary_type_info  
}
```

Definition at line 184 of file ng_bridge.c.

7.82.4.8 struct `ng_parse_array_info ng_bridge_hary_type_info` [static]**Initial value:**

```
{
    &ng_bridge_host_type,
    ng_bridge_getTableLength
}
```

Definition at line 180 of file `ng_bridge.c`.

7.82.4.9 struct `ng_parse_type ng_bridge_host_ary_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bridge_host_ary_type_fields
}
```

Definition at line 190 of file `ng_bridge.c`.

7.82.4.10 struct `ng_parse_struct_field ng_bridge_host_ary_type_fields[] = NG_BRIDGE_HOST_ARY_TYPE_INFO(&ng_bridge_hary_type)` [static]

Definition at line 189 of file `ng_bridge.c`.

7.82.4.11 struct `ng_parse_type ng_bridge_host_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_bridge_host_type_fields
}
```

Definition at line 176 of file `ng_bridge.c`.

7.82.4.12 struct `ng_parse_struct_field ng_bridge_host_type_fields[] = NG_BRIDGE_HOST_TYPE_INFO(&ng_parse_enaddr_type)` [static]

Definition at line 175 of file `ng_bridge.c`.

7.82.4.13 struct `ng_parse_type ng_bridge_ipfwary_type` [static]**Initial value:**

```
{
    &ng_parse_fixedarray_type,
    &ng_bridge_ipfwary_type_info
}
```

Definition at line 200 of file `ng_bridge.c`.

7.82.4.14 `struct ng_parse_fixedarray_info ng_bridge_ipfwary_type_info` [static]

Initial value:

```
{
    &ng_parse_uint8_type,
    NG_BRIDGE_MAX_LINKS
}
```

Definition at line 196 of file ng_bridge.c.

7.82.4.15 `ng_newhook_t ng_bridge_newhook` [static]

Definition at line 122 of file ng_bridge.c.

7.82.4.16 `ng_rcvdata_t ng_bridge_rcvdata` [static]

Definition at line 123 of file ng_bridge.c.

7.82.4.17 `ng_rcvmsg_t ng_bridge_rcvmsg` [static]

Definition at line 120 of file ng_bridge.c.

7.82.4.18 `ng_shutdown_t ng_bridge_shutdown` [static]

Definition at line 121 of file ng_bridge.c.

7.82.4.19 `struct ng_parse_type ng_bridge_stats_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_bridge_stats_type_fields
}
```

Definition at line 214 of file ng_bridge.c.

7.82.4.20 `struct ng_parse_struct_field ng_bridge_stats_type_fields[] =
 NG_BRIDGE_STATS_TYPE_INFO` [static]

Definition at line 213 of file ng_bridge.c.

7.82.4.21 `struct ng_type ng_bridge_typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_BRIDGE_NODE_TYPE,
    .constructor =  ng_bridge_constructor,
    .rcvmsg =       ng_bridge_rcvmsg,
    .shutdown =     ng_bridge_shutdown,
    .newhook =      ng_bridge_newhook,
    .rcvdata =      ng_bridge_rcvdata,
    .disconnect =   ng_bridge_disconnect,
    .cmdlist =      ng_bridge_cmdlist,
}
```

Definition at line 274 of file ng_bridge.c.

7.83 /usr/src/sys/netgraph/ng_bridge.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_bridge_config](#)
- struct [ng_bridge_link_stats](#)
- struct [ng_bridge_host](#)
- struct [ng_bridge_host_ary](#)

Defines

- #define [NG_BRIDGE_NODE_TYPE](#) "bridge"
- #define [NGM_BRIDGE_COOKIE](#) 967239368
- #define [NG_BRIDGE_HOOK_LINK_PREFIX](#) "link"
- #define [NG_BRIDGE_HOOK_LINK_FMT](#) "link%d"
- #define [NG_BRIDGE_MAX_LINKS](#) 32
- #define [NG_BRIDGE_CONFIG_TYPE_INFO](#)(ainfo)
- #define [NG_BRIDGE_STATS_TYPE_INFO](#)
- #define [NG_BRIDGE_HOST_TYPE_INFO](#)(entype)
- #define [NG_BRIDGE_HOST_ARY_TYPE_INFO](#)(harytype)

Enumerations

- enum {
 - [NGM_BRIDGE_SET_CONFIG](#) = 1, [NGM_BRIDGE_GET_CONFIG](#), [NGM_BRIDGE_RESET](#),
 - [NGM_BRIDGE_GET_STATS](#),
 - [NGM_BRIDGE_CLR_STATS](#), [NGM_BRIDGE_GETCLR_STATS](#), [NGM_BRIDGE_GET_](#)
 - [TABLE](#) }

7.83.1 Define Documentation

7.83.1.1 #define [NG_BRIDGE_CONFIG_TYPE_INFO](#)(ainfo)

Value:

```

{
    \
    { "ipfw",          (ainfo)          }, \
    { "debugLevel",   &ng_parse_uint8_type }, \
    { "loopTimeout",  &ng_parse_uint32_type }, \
    { "maxStaleness", &ng_parse_uint32_type }, \
    { "minStableAge", &ng_parse_uint32_type }, \
    { NULL }
}
  
```

Definition at line 67 of file [ng_bridge.h](#).

7.83.1.2 #define NG_BRIDGE_HOOK_LINK_FMT "link%d"

Definition at line 52 of file ng_bridge.h.

7.83.1.3 #define NG_BRIDGE_HOOK_LINK_PREFIX "link"

Definition at line 51 of file ng_bridge.h.

Referenced by ng_bridge_newhook().

7.83.1.4 #define NG_BRIDGE_HOST_ARY_TYPE_INFO(harytype)**Value:**

```
{
    \
    { "numHosts",      &ng_parse_uint32_type  }, \
    { "hosts",        (harytype)             }, \
    { NULL }
}
```

Definition at line 137 of file ng_bridge.h.

7.83.1.5 #define NG_BRIDGE_HOST_TYPE_INFO(entype)**Value:**

```
{
    \
    { "addr",          (entype)                }, \
    { "linkNum",      &ng_parse_uint16_type  }, \
    { "age",          &ng_parse_uint16_type  }, \
    { "staleness",    &ng_parse_uint16_type  }, \
    { NULL }
}
```

Definition at line 122 of file ng_bridge.h.

7.83.1.6 #define NG_BRIDGE_MAX_LINKS 32

Definition at line 55 of file ng_bridge.h.

Referenced by ng_bridge_disconnect(), ng_bridge_newhook(), ng_bridge_rcvdata(), ng_bridge_rcvmsg(), and ng_bridge_timeout().

7.83.1.7 #define NG_BRIDGE_NODE_TYPE "bridge"

Definition at line 47 of file ng_bridge.h.

7.83.1.8 #define NG_BRIDGE_STATS_TYPE_INFO**Value:**

```

{
    { "recvOctets",      &ng_parse_uint64_type }, \
    { "recvPackets",   &ng_parse_uint64_type }, \
    { "recvMulticast", &ng_parse_uint64_type }, \
    { "recvBroadcast", &ng_parse_uint64_type }, \
    { "recvUnknown",   &ng_parse_uint64_type }, \
    { "recvRunts",     &ng_parse_uint64_type }, \
    { "recvInvalid",   &ng_parse_uint64_type }, \
    { "xmitOctets",    &ng_parse_uint64_type }, \
    { "xmitPackets",   &ng_parse_uint64_type }, \
    { "xmitMulticasts", &ng_parse_uint64_type }, \
    { "xmitBroadcasts", &ng_parse_uint64_type }, \
    { "loopDrops",     &ng_parse_uint64_type }, \
    { "loopDetects",   &ng_parse_uint64_type }, \
    { "memoryFailures", &ng_parse_uint64_type }, \
    { NULL }
}

```

Definition at line 95 of file ng_bridge.h.

7.83.1.9 #define NGM_BRIDGE_COOKIE 967239368

Definition at line 48 of file ng_bridge.h.

Referenced by ng_bridge_rcvmsg().

7.83.2 Enumeration Type Documentation

7.83.2.1 anonymous enum

Enumerator:

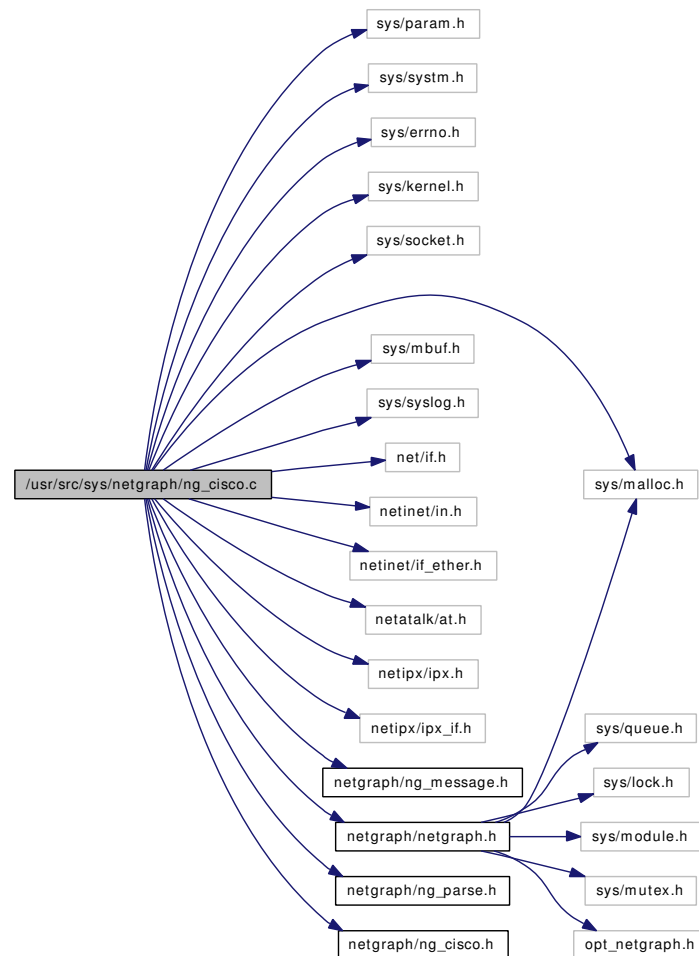
NGM_BRIDGE_SET_CONFIG
NGM_BRIDGE_GET_CONFIG
NGM_BRIDGE_RESET
NGM_BRIDGE_GET_STATS
NGM_BRIDGE_CLR_STATS
NGM_BRIDGE_GETCLR_STATS
NGM_BRIDGE_GET_TABLE

Definition at line 144 of file ng_bridge.h.

7.84 /usr/src/sys/netgraph/ng_cisco.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/socket.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet/if_ether.h>
#include <netatalk/at.h>
#include <netipx/ipx.h>
#include <netipx/ipx_if.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_cisco.h>
```

Include dependency graph for ng_cisco.c:



Data Structures

- struct [cisco_header](#)
- struct [cisco_packet](#)
- struct [protoent](#)
- struct [cisco_priv](#)

Defines

- #define [CISCO_MULTICAST](#) 0x8f
- #define [CISCO_UNICAST](#) 0x0f
- #define [CISCO_KEEPA_LIVE](#) 0x8035
- #define [CISCO_ADDR_REQ](#) 0
- #define [CISCO_ADDR_REPLY](#) 1
- #define [CISCO_KEEPA_LIVE_REQ](#) 2
- #define [KEEPA_LIVE_SECS](#) 10
- #define [CISCO_HEADER_LEN](#) sizeof (struct [cisco_header](#))
- #define [CISCO_PACKET_LEN](#) (sizeof(struct [cisco_packet](#)))

Typedefs

- typedef [cisco_priv](#) * [sc_p](#)

Functions

- static int [cisco_input](#) ([sc_p](#) sc, [item_p](#) item)
- static void [cisco_keeplive](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static int [cisco_send](#) ([sc_p](#) sc, int type, long par1, long par2)
- static void [cisco_notify](#) ([sc_p](#) sc, [uint32_t](#) cmd)
- [NETGRAPH_INIT](#) (cisco,&typestruct)
- static int [cisco_constructor](#) ([node_p](#) node)
- static int [cisco_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [cisco_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [cisco_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [cisco_shutdown](#) ([node_p](#) node)
- static int [cisco_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [cisco_constructor](#)
- static [ng_rcvmsg_t](#) [cisco_rcvmsg](#)
- static [ng_shutdown_t](#) [cisco_shutdown](#)
- static [ng_newhook_t](#) [cisco_newhook](#)
- static [ng_rcvdata_t](#) [cisco_rcvdata](#)
- static [ng_disconnect_t](#) [cisco_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_cisco_ipaddr_type_fields](#) [] = [NG_CISCO_IPADDR_TYPE_](#)
[INFO](#)
- static struct [ng_parse_type](#) [ng_cisco_ipaddr_type](#)
- static struct [ng_parse_struct_field](#) [ng_cisco_stats_type_fields](#) [] = [NG_CISCO_STATS_TYPE_](#)
[INFO](#)
- static struct [ng_parse_type](#) [ng_cisco_stats_type](#)
- static struct [ng_cmdlist](#) [ng_cisco_cmdlist](#) []
- static struct [ng_type](#) typestruct

7.84.1 Define Documentation

7.84.1.1 #define CISCO_ADDR_REPLY 1

Definition at line 72 of file [ng_cisco.c](#).

Referenced by [cisco_input](#)().

7.84.1.2 #define CISCO_ADDR_REQ 0

Definition at line 71 of file [ng_cisco.c](#).

Referenced by [cisco_input](#)().

7.84.1.3 #define CISCO_HEADER_LEN sizeof (struct [cisco_header](#))

Definition at line 83 of file `ng_cisco.c`.

Referenced by `cisco_rcvdata()`, and `cisco_send()`.

7.84.1.4 #define CISCO_KEEPALIVE 0x8035

Definition at line 70 of file `ng_cisco.c`.

Referenced by `cisco_input()`, and `cisco_send()`.

7.84.1.5 #define CISCO_KEEPALIVE_REQ 2

Definition at line 73 of file `ng_cisco.c`.

Referenced by `cisco_input()`, and `cisco_keepalive()`.

7.84.1.6 #define CISCO_MULTICAST 0x8f

Definition at line 68 of file `ng_cisco.c`.

Referenced by `cisco_input()`, and `cisco_send()`.

7.84.1.7 #define CISCO_PACKET_LEN (sizeof(struct [cisco_packet](#)))

Definition at line 94 of file `ng_cisco.c`.

Referenced by `cisco_send()`.

7.84.1.8 #define CISCO_UNICAST 0x0f

Definition at line 69 of file `ng_cisco.c`.

Referenced by `cisco_input()`, and `cisco_rcvdata()`.

7.84.1.9 #define KEEPALIVE_SECS 10

Definition at line 75 of file `ng_cisco.c`.

Referenced by `cisco_keepalive()`, `cisco_newhook()`, and `cisco_rcvmsg()`.

7.84.2 Typedef Documentation**7.84.2.1 typedef struct [cisco_priv](#)* [sc_p](#)**

Definition at line 116 of file `ng_cisco.c`.

7.84.3 Function Documentation

7.84.3.1 static int cisco_constructor (node_p node) [static]

Definition at line 192 of file ng_cisco.c.

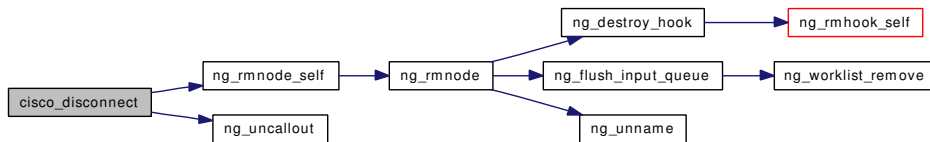
References ng_callout_init, and NG_NODE_SET_PRIVATE.

7.84.3.2 static int cisco_disconnect (hook_p hook) [static]

Definition at line 418 of file ng_cisco.c.

References protoent::af, protoent::hook, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_IS_INVALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), and ng_uncallout().

Here is the call graph for this function:



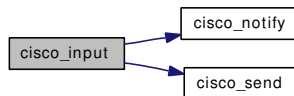
7.84.3.3 static int cisco_input (sc_p sc, item_p item) [static]

Definition at line 442 of file ng_cisco.c.

References CISCO_ADDR_REPLY, CISCO_ADDR_REQ, CISCO_KEEPALIVE, CISCO_KEEPALIVE_REQ, CISCO_MULTICAST, cisco_notify(), cisco_send(), CISCO_UNICAST, protoent::hook, NG_FREE_ITEM, NG_FWD_NEW_DATA, NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGI_M, NGM_CISCO_COOKIE, NGM_CISCO_GET_IPADDR, NGM_LINK_IS_UP, and ng_async_private::node.

Referenced by cisco_rcvdata().

Here is the call graph for this function:



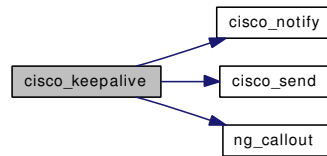
7.84.3.4 static void cisco_keepalive (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 583 of file ng_cisco.c.

References CISCO_KEEPALIVE_REQ, cisco_notify(), cisco_send(), KEEPALIVE_SECS, ng_callout(), and NGM_LINK_IS_DOWN.

Referenced by cisco_newhook().

Here is the call graph for this function:

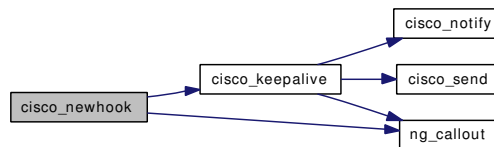


7.84.3.5 static int cisco_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 217 of file ng_cisco.c.

References cisco_keepalive(), KEEPALIVE_SECS, ng_callout(), NG_CISCO_HOOK_APPLETALK, NG_CISCO_HOOK_DEBUG, NG_CISCO_HOOK_DOWNSTREAM, NG_CISCO_HOOK_INET, NG_CISCO_HOOK_IPX, NG_HOOK_SET_PRIVATE, and NG_NODE_PRIVATE.

Here is the call graph for this function:



7.84.3.6 static void cisco_notify (sc_p sc, uint32_t cmd) [static]

Definition at line 638 of file ng_cisco.c.

References NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGM_FLOW_COOKIE, and ng_async_private::node.

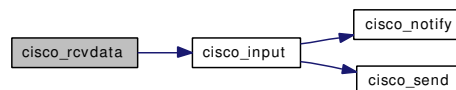
Referenced by cisco_input(), and cisco_keepalive().

7.84.3.7 static int cisco_rcvdata (hook_p hook, item_p item) [static]

Definition at line 344 of file ng_cisco.c.

References cisco_header::address, protoent::af, CISCO_HEADER_LEN, cisco_input(), CISCO_UNICAST, cisco_header::control, NG_FREE_ITEM, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, NGI_M, and cisco_header::protocol.

Here is the call graph for this function:



7.84.3.8 static int cisco_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 248 of file ng_cisco.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, `KEEPALIVE_SECS`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NG_TEXTRESPONSE`, `NGF_RESP`, `NGI_GET_MSG`, `NGM_CISCO_COOKIE`, `NGM_CISCO_GET_IPADDR`, `NGM_CISCO_GET_STATUS`, `NGM_CISCO_SET_IPADDR`, `NGM_GENERIC_COOKIE`, `NGM_TEXT_STATUS`, and `ng_mesg::ng_msghdr::typecookie`.

7.84.3.9 `static int cisco_send` (`sc_p sc`, `int type`, `long par1`, `long par2`) [static]

Definition at line 598 of file `ng_cisco.c`.

References `cisco_header::address`, `CISCO_HEADER_LEN`, `CISCO_KEEPALIVE`, `CISCO_MULTICAST`, `CISCO_PACKET_LEN`, `cisco_header::control`, `NG_SEND_DATA_ONLY`, `cisco_packet::par1`, `cisco_packet::par2`, `cisco_header::protocol`, `cisco_packet::rel`, `cisco_packet::time0`, `cisco_packet::time1`, and `cisco_packet::type`.

Referenced by `cisco_input()`, and `cisco_keepalive()`.

7.84.3.10 `static int cisco_shutdown` (`node_p node`) [static]

Definition at line 402 of file `ng_cisco.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `ng_async_private::node`.

7.84.3.11 `NETGRAPH_INIT` (`cisco`, `& typestruct`)

7.84.4 Variable Documentation

7.84.4.1 `ng_constructor_t cisco_constructor` [static]

Definition at line 119 of file `ng_cisco.c`.

7.84.4.2 `ng_disconnect_t cisco_disconnect` [static]

Definition at line 124 of file `ng_cisco.c`.

7.84.4.3 `ng_newhook_t cisco_newhook` [static]

Definition at line 122 of file `ng_cisco.c`.

7.84.4.4 `ng_rcvdata_t cisco_rcvdata` [static]

Definition at line 123 of file `ng_cisco.c`.

7.84.4.5 `ng_rcvmsg_t cisco_rcvmsg` [static]

Definition at line 120 of file `ng_cisco.c`.

7.84.4.6 `ng_shutdown_t` `cisco_shutdown` [static]

Definition at line 121 of file `ng_cisco.c`.

7.84.4.7 `struct ng_cmdlist` `ng_cisco_cmdlist[]` [static]**Initial value:**

```
{
    {
        NGM_CISCO_COOKIE,
        NGM_CISCO_SET_IPADDR,
        "setipaddr",
        &ng_cisco_ipaddr_type,
        NULL
    },
    {
        NGM_CISCO_COOKIE,
        NGM_CISCO_GET_IPADDR,
        "getipaddr",
        NULL,
        &ng_cisco_ipaddr_type
    },
    {
        NGM_CISCO_COOKIE,
        NGM_CISCO_GET_STATUS,
        "getstats",
        NULL,
        &ng_cisco_stats_type
    },
    { 0 }
}
```

Definition at line 149 of file `ng_cisco.c`.

7.84.4.8 `struct ng_parse_type` `ng_cisco_ipaddr_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_cisco_ipaddr_type_fields
}
```

Definition at line 135 of file `ng_cisco.c`.

7.84.4.9 `struct ng_parse_struct_field` `ng_cisco_ipaddr_type_fields[]` =
`NG_CISCO_IPADDR_TYPE_INFO` [static]

Definition at line 134 of file `ng_cisco.c`.

7.84.4.10 `struct ng_parse_type` `ng_cisco_stats_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_cisco_stats_type_fields
}
```

Definition at line 143 of file ng_cisco.c.

7.84.4.11 `struct ng_parse_struct_field ng_cisco_stats_type_fields[] =`
`NG_CISCO_STATS_TYPE_INFO [static]`

Definition at line 142 of file ng_cisco.c.

7.84.4.12 `struct ng_type typestruct [static]`

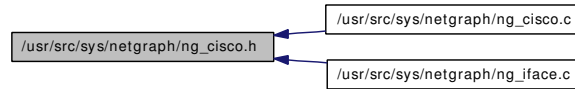
Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_CISCO_NODE_TYPE,
    .constructor =  cisco_constructor,
    .rcvmsg =      cisco_rcvmsg,
    .shutdown =    cisco_shutdown,
    .newhook =     cisco_newhook,
    .rcvdata =     cisco_rcvdata,
    .disconnect =  cisco_disconnect,
    .cmdlist =     ng_cisco_cmdlist,
}
```

Definition at line 175 of file ng_cisco.c.

7.85 /usr/src/sys/netgraph/ng_cisco.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_cisco_ipaddr](#)
- struct [ng_cisco_stats](#)

Defines

- #define [NG_CISCO_NODE_TYPE](#) "cisco"
- #define [NGM_CISCO_COOKIE](#) 860707227
- #define [NG_CISCO_HOOK_DOWNSTREAM](#) "downstream"
- #define [NG_CISCO_HOOK_INET](#) "inet"
- #define [NG_CISCO_HOOK_INET6](#) "inet6"
- #define [NG_CISCO_HOOK_APPLETALK](#) "atalk"
- #define [NG_CISCO_HOOK_IPX](#) "ipx"
- #define [NG_CISCO_HOOK_DEBUG](#) "debug"
- #define [NG_CISCO_IPADDR_TYPE_INFO](#)
- #define [NG_CISCO_STATS_TYPE_INFO](#)

Enumerations

- enum { [NGM_CISCO_SET_IPADDR](#) = 1, [NGM_CISCO_GET_IPADDR](#), [NGM_CISCO_GET_STATUS](#) }

7.85.1 Define Documentation

7.85.1.1 #define [NG_CISCO_HOOK_APPLETALK](#) "atalk"

Definition at line 55 of file [ng_cisco.h](#).

Referenced by [cisco_newhook\(\)](#).

7.85.1.2 #define [NG_CISCO_HOOK_DEBUG](#) "debug"

Definition at line 57 of file [ng_cisco.h](#).

Referenced by [cisco_newhook\(\)](#).

7.85.1.3 #define [NG_CISCO_HOOK_DOWNSTREAM](#) "downstream"

Definition at line 52 of file [ng_cisco.h](#).

Referenced by [cisco_newhook\(\)](#).

7.85.1.4 #define NG_CISCO_HOOK_INET "inet"

Definition at line 53 of file ng_cisco.h.

Referenced by cisco_newhook().

7.85.1.5 #define NG_CISCO_HOOK_INET6 "inet6"

Definition at line 54 of file ng_cisco.h.

7.85.1.6 #define NG_CISCO_HOOK_IPX "ipx"

Definition at line 56 of file ng_cisco.h.

Referenced by cisco_newhook().

7.85.1.7 #define NG_CISCO_IPADDR_TYPE_INFO

Value:

```
{
    { "ipaddr",          &ng_parse_ipaddr_type }, \
    { "netmask",        &ng_parse_ipaddr_type }, \
    { NULL }
}
```

Definition at line 72 of file ng_cisco.h.

7.85.1.8 #define NG_CISCO_NODE_TYPE "cisco"

Definition at line 48 of file ng_cisco.h.

7.85.1.9 #define NG_CISCO_STATS_TYPE_INFO

Value:

```
{
    { "seqRetries",     &ng_parse_uint32_type }, \
    { "keepAlivePeriod", &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 84 of file ng_cisco.h.

7.85.1.10 #define NGM_CISCO_COOKIE 860707227

Definition at line 49 of file ng_cisco.h.

Referenced by cisco_input(), cisco_rcvmsg(), and ng_iface_rcvmsg().

7.85.2 Enumeration Type Documentation

7.85.2.1 anonymous enum

Enumerator:

NGM_CISCO_SET_IPADDR

NGM_CISCO_GET_IPADDR

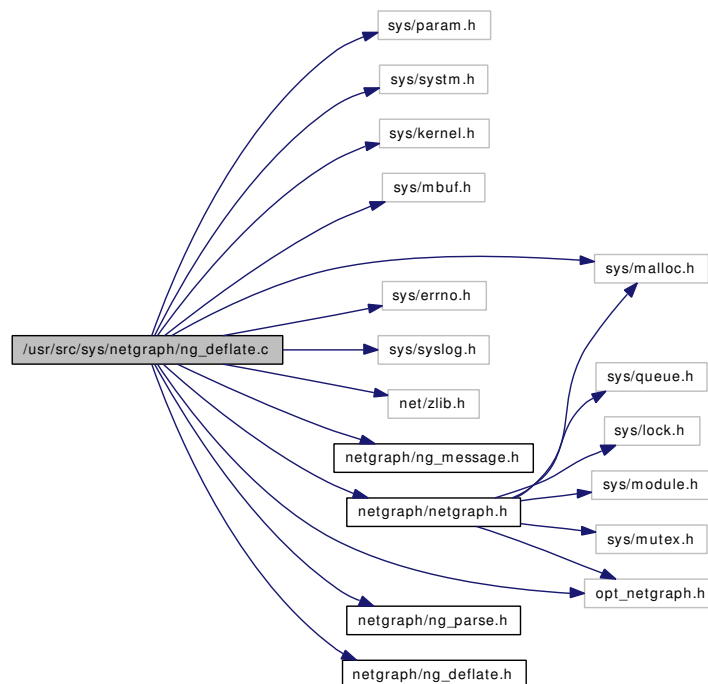
NGM_CISCO_GET_STATUS

Definition at line 60 of file ng_cisco.h.

7.86 /usr/src/sys/netgraph/ng_deflate.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <net/zlib.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_deflate.h>
#include "opt_netgraph.h"
```

Include dependency graph for ng_deflate.c:



Data Structures

- struct [ng_deflate_private](#)

Defines

- #define [DEFLATE_HDRLEN](#) 2
- #define [PROT_COMPD](#) 0x00fd
- #define [DEFLATE_BUF_SIZE](#) 4096
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef [ng_deflate_private](#) * [priv_p](#)

Functions

- [MALLOC_DEFINE](#) (M_NETGRAPH_DEFLATE,"netgraph_deflate","netgraph deflate node ")
- static void * [z_alloc](#) (void *, u_int items, u_int size)
- static void [z_free](#) (void *, void *ptr)
- static int [ng_deflate_compress](#) ([node_p](#) node, struct mbuf *m, struct mbuf **resultp)
- static int [ng_deflate_decompress](#) ([node_p](#) node, struct mbuf *m, struct mbuf **resultp)
- static void [ng_deflate_reset_req](#) ([node_p](#) node)
- [NETGRAPH_INIT](#) (deflate,&[ng_deflate_tpestruct](#))
- [MODULE_DEPEND](#) (ng_deflate, zlib, 1, 1, 1)
- static int [ng_deflate_constructor](#) ([node_p](#) node)
- static int [ng_deflate_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_deflate_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_deflate_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_deflate_shutdown](#) ([node_p](#) node)
- static int [ng_deflate_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_deflate_constructor](#)
- static [ng_rcvmsg_t](#) [ng_deflate_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_deflate_shutdown](#)
- static [ng_newhook_t](#) [ng_deflate_newhook](#)
- static [ng_rcvdata_t](#) [ng_deflate_rcvdata](#)
- static [ng_disconnect_t](#) [ng_deflate_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_deflate_config_type_fields](#) [] = [NG_DEFLATE_CONFIG_INFO](#)
- static struct [ng_parse_type](#) [ng_deflate_config_type](#)
- static struct [ng_parse_struct_field](#) [ng_deflate_stats_type_fields](#) [] = [NG_DEFLATE_STATS_INFO](#)
- static struct [ng_parse_type](#) [ng_deflate_stat_type](#)
- static struct [ng_cmdlist](#) [ng_deflate_cmds](#) []
- static struct [ng_type](#) [ng_deflate_tpestruct](#)

7.86.1 Define Documentation

7.86.1.1 #define DEFLATE_BUF_SIZE 4096

Definition at line 58 of file [ng_deflate.c](#).

Referenced by [ng_deflate_compress\(\)](#), and [ng_deflate_decompress\(\)](#).

7.86.1.2 #define DEFLATE_HDRLLEN 2

Definition at line 54 of file ng_deflate.c.

Referenced by ng_deflate_compress().

7.86.1.3 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 163 of file ng_deflate.c.

7.86.1.4 #define PROT_COMPD 0x00fd

Definition at line 56 of file ng_deflate.c.

Referenced by ng_deflate_compress(), ng_deflate_decompress(), ng_ppp_comp_rcv(), ng_ppp_comp_xmit(), and ng_ppp_rcvdata_compress().

7.86.2 Typedef Documentation**7.86.2.1 typedef struct ng_deflate_private* priv_p**

Definition at line 71 of file ng_deflate.c.

7.86.3 Function Documentation**7.86.3.1 MALLOC_DEFINE (M_NETGRAPH_DEFLATE, "netgraph_deflate", "netgraph deflate node ")****7.86.3.2 MODULE_DEPEND (ng_deflate, zlib, 1, 1, 1)****7.86.3.3 NETGRAPH_INIT (deflate, & ng_deflate_tpestruct)****7.86.3.4 static int ng_deflate_compress (node_p node, struct mbuf * m, struct mbuf ** resultp) [static]**

Definition at line 442 of file ng_deflate.c.

References DEFLATE_BUF_SIZE, DEFLATE_HDRLLEN, NG_FREE_M, NG_NODE_PRIVATE, and PROT_COMPD.

Referenced by ng_deflate_rcvdata().

7.86.3.5 static int ng_deflate_constructor (node_p node) [static]

Definition at line 173 of file ng_deflate.c.

References NG_NODE_FORCE_WRITER, and NG_NODE_SET_PRIVATE.

7.86.3.6 static int ng_deflate_decompress (node_p node, struct mbuf * m, struct mbuf ** resultp) [static]

Definition at line 528 of file ng_deflate.c.

References DEFLATE_BUF_SIZE, NG_FREE_M, NG_NODE_PRIVATE, and PROT_COMPD.

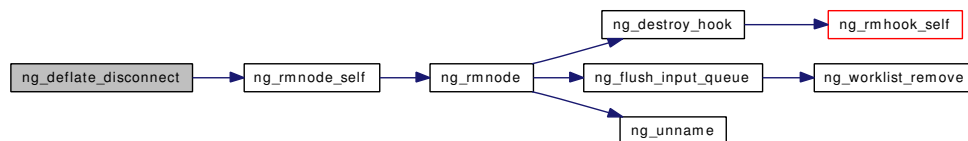
Referenced by ng_deflate_rcvdata().

7.86.3.7 static int ng_deflate_disconnect (hook_p hook) [static]

Definition at line 396 of file ng_deflate.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.86.3.8 static int ng_deflate_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 193 of file ng_deflate.c.

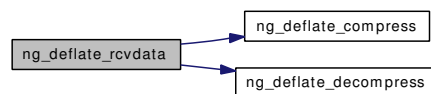
References NG_DEFLATE_HOOK_COMP, NG_DEFLATE_HOOK_DECOMP, NG_NODE_NUMHOOKS, and NG_NODE_PRIVATE.

7.86.3.9 static int ng_deflate_rcvdata (hook_p hook, item_p item) [static]

Definition at line 326 of file ng_deflate.c.

References ng_deflate_compress(), ng_deflate_decompress(), NG_FREE_ITEM, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_MKMESSAGE, NG_NODE_PRIVATE, NG_SEND_MSG_ID, NGI_GET_M, NGM_DEFLATE_COOKIE, and NGM_DEFLATE_RESETRREQ.

Here is the call graph for this function:

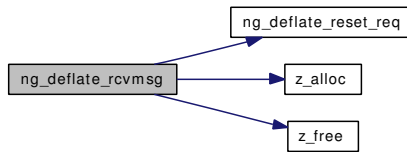


7.86.3.10 static int ng_deflate_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 214 of file ng_deflate.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERRROUT, ng_mesg::header, ng_deflate_reset_req(), NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGI_RETADDR, NGM_DEFLATE_CLR_STATS, NGM_DEFLATE_CONFIG, NGM_DEFLATE_COOKIE, NGM_DEFLATE_GET_STATS, NGM_DEFLATE_GETCLR_STATS, NGM_DEFLATE_RESETRREQ, ng_mesg::ng_msghdr::typecookie, z_alloc(), and z_free().

Here is the call graph for this function:



7.86.3.11 static void `ng_deflate_reset_req` (`node_p node`) [static]

Definition at line 672 of file `ng_deflate.c`.

References `NG_NODE_PRIVATE`.

Referenced by `ng_deflate_rcvmsg()`.

7.86.3.12 static int `ng_deflate_shutdown` (`node_p node`) [static]

Definition at line 374 of file `ng_deflate.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.86.3.13 static void * `z_alloc` (`void *`, `u_int items`, `u_int size`) [static]

Definition at line 424 of file `ng_deflate.c`.

Referenced by `ng_deflate_rcvmsg()`.

7.86.3.14 static void `z_free` (`void *`, `void * ptr`) [static]

Definition at line 431 of file `ng_deflate.c`.

Referenced by `ng_deflate_rcvmsg()`.

7.86.4 Variable Documentation

7.86.4.1 struct `ng_cmdlist ng_deflate_cmds`[] [static]

Definition at line 107 of file `ng_deflate.c`.

7.86.4.2 struct `ng_parse_type ng_deflate_config_type` [static]

Initial value:

```

{
    &ng_parse_struct_type,
    ng_deflate_config_type_fields
}
  
```

Definition at line 93 of file `ng_deflate.c`.

7.86.4.3 `struct ng_parse_struct_field ng_deflate_config_type_fields[] = NG_DEFLATE_CONFIG_INFO` [static]

Definition at line 92 of file ng_deflate.c.

7.86.4.4 `ng_constructor_t ng_deflate_constructor` [static]

Definition at line 74 of file ng_deflate.c.

7.86.4.5 `ng_disconnect_t ng_deflate_disconnect` [static]

Definition at line 79 of file ng_deflate.c.

7.86.4.6 `ng_newhook_t ng_deflate_newhook` [static]

Definition at line 77 of file ng_deflate.c.

7.86.4.7 `ng_rcvdata_t ng_deflate_rcvdata` [static]

Definition at line 78 of file ng_deflate.c.

7.86.4.8 `ng_rcvmsg_t ng_deflate_rcvmsg` [static]

Definition at line 75 of file ng_deflate.c.

7.86.4.9 `ng_shutdown_t ng_deflate_shutdown` [static]

Definition at line 76 of file ng_deflate.c.

7.86.4.10 `struct ng_parse_type ng_deflate_stat_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_deflate_stats_type_fields
}
```

Definition at line 101 of file ng_deflate.c.

7.86.4.11 `struct ng_parse_struct_field ng_deflate_stats_type_fields[] = NG_DEFLATE_STATS_INFO` [static]

Definition at line 100 of file ng_deflate.c.

7.86.4.12 struct `ng_type ng_deflate_t`struct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_DEFLATE_NODE_TYPE,
    .constructor =  ng_deflate_constructor,
    .rcvmsg =       ng_deflate_rcvmsg,
    .shutdown =     ng_deflate_shutdown,
    .newhook =      ng_deflate_newhook,
    .rcvdata =      ng_deflate_rcvdata,
    .disconnect =   ng_deflate_disconnect,
    .cmdlist =      ng_deflate_cmds,
}
```

Definition at line 147 of file `ng_deflate.c`.

7.87 /usr/src/sys/netgraph/ng_deflate.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_deflate_config](#)
- struct [ng_deflate_stats](#)

Defines

- #define [NG_DEFLATE_NODE_TYPE](#) "deflate"
- #define [NGM_DEFLATE_COOKIE](#) 1166642656
- #define [NG_DEFLATE_HOOK_COMP](#) "comp"
- #define [NG_DEFLATE_HOOK_DECOMP](#) "decomp"
- #define [NG_DEFLATE_CONFIG_INFO](#)
- #define [NG_DEFLATE_STATS_INFO](#)

Enumerations

- enum {
 - [NGM_DEFLATE_CONFIG](#) = 1, [NGM_DEFLATE_RESETREQ](#), [NGM_DEFLATE_GET_STATS](#),
 - [NGM_DEFLATE_CLR_STATS](#),
 - [NGM_DEFLATE_GETCLR_STATS](#) }

7.87.1 Define Documentation

7.87.1.1 #define NG_DEFLATE_CONFIG_INFO

Value:

```

{
    { "enable",      \
      &ng_parse_uint8_type }, \
    { "windowBits", &ng_parse_uint8_type }, \
    { NULL }
}
  
```

Definition at line 48 of file `ng_deflate.h`.

7.87.1.2 #define NG_DEFLATE_HOOK_COMP "comp"

Definition at line 38 of file `ng_deflate.h`.

Referenced by `ng_deflate_newhook()`.

7.87.1.3 #define NG_DEFLATE_HOOK_DECOMP "decomp"

Definition at line 39 of file ng_deflate.h.

Referenced by ng_deflate_newhook().

7.87.1.4 #define NG_DEFLATE_NODE_TYPE "deflate"

Definition at line 34 of file ng_deflate.h.

7.87.1.5 #define NG_DEFLATE_STATS_INFO

Value:

```
{
    { "FramesPlain", &ng_parse_uint64_type }, \
    { "FramesComp", &ng_parse_uint64_type }, \
    { "FramesUncomp", &ng_parse_uint64_type }, \
    { "InOctets", &ng_parse_uint64_type }, \
    { "OutOctets", &ng_parse_uint64_type }, \
    { "Errors", &ng_parse_uint64_type }, \
    { NULL }
}
```

Definition at line 65 of file ng_deflate.h.

7.87.1.6 #define NGM_DEFLATE_COOKIE 1166642656

Definition at line 35 of file ng_deflate.h.

Referenced by ng_deflate_rcvdata(), and ng_deflate_rcvmsg().

7.87.2 Enumeration Type Documentation**7.87.2.1 anonymous enum**

Enumerator:

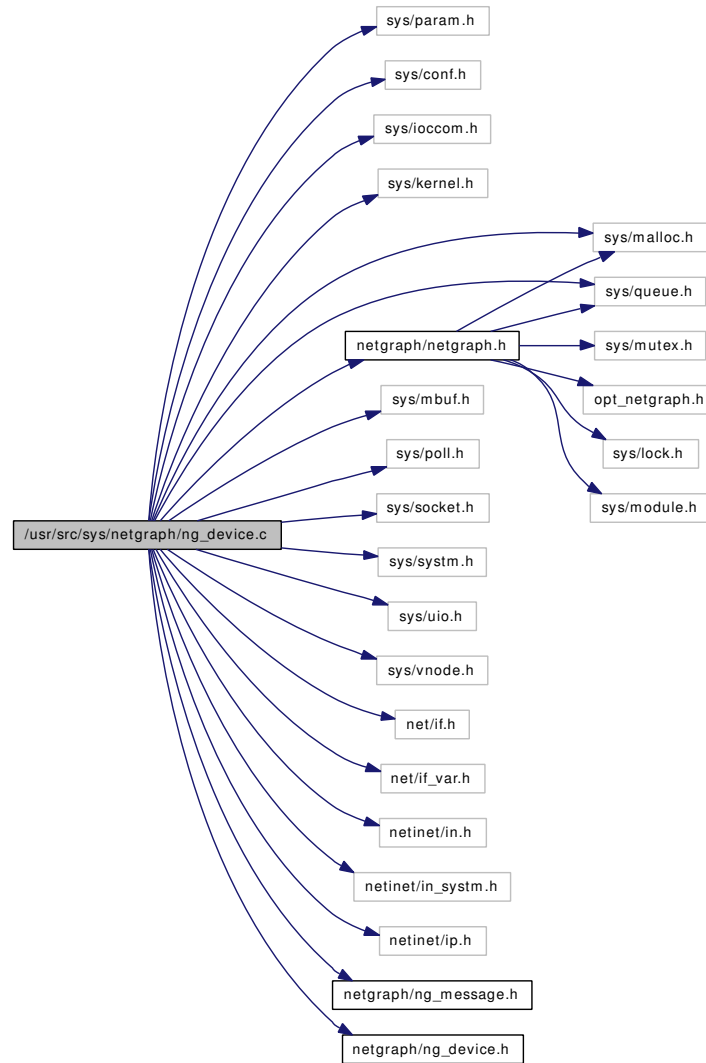
```
NGM_DEFLATE_CONFIG
NGM_DEFLATE_RESETREQ
NGM_DEFLATE_GET_STATS
NGM_DEFLATE_CLR_STATS
NGM_DEFLATE_GETCLR_STATS
```

Definition at line 76 of file ng_deflate.h.

7.88 /usr/src/sys/netgraph/ng_device.c File Reference

```
#include <sys/param.h>
#include <sys/conf.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/poll.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/system.h>
#include <sys/uio.h>
#include <sys/vnode.h>
#include <net/if.h>
#include <net/if_var.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_device.h>
```

Include dependency graph for ng_device.c:



Data Structures

- struct [ngd_private](#)

Defines

- #define [DBG](#) do { } while (0)
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)
- #define [NGDF_OPEN](#) 0x0001
- #define [NGDF_RWAIT](#) 0x0002
- #define [MAX_NGD](#) 999

Typedefs

- typedef [ngd_private](#) * [priv_p](#)

Functions

- static int `ng_device_mod_event` (module_t, int, void *)
- `NETGRAPH_INIT` (device,&ngd_tpestruct)
- static int `ng_device_constructor` (node_p node)
- static int `ng_device_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_device_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_device_rcvdata` (hook_p hook, item_p item)
- static int `ng_device_disconnect` (hook_p hook)
- static int `ng_device_shutdown` (node_p node)
- static int `ngdopen` (struct cdev *dev, int flag, int mode, struct thread *td)
- static int `ngdclose` (struct cdev *dev, int flag, int mode, struct thread *td)
- static int `ngdread` (struct cdev *dev, struct uio *uio, int flag)
- static int `ngdwrite` (struct cdev *dev, struct uio *uio, int flag)
- static int `ngdpoll` (struct cdev *dev, int events, struct thread *td)

Variables

- static `ng_constructor_t` `ng_device_constructor`
- static `ng_rcvmsg_t` `ng_device_rcvmsg`
- static `ng_shutdown_t` `ng_device_shutdown`
- static `ng_newhook_t` `ng_device_newhook`
- static `ng_rcvdata_t` `ng_device_rcvdata`
- static `ng_disconnect_t` `ng_device_disconnect`
- static struct `ng_type` `ngd_tpestruct`
- static struct `unrhdr *` `ngd_unit`
- static `d_close_t` `ngdclose`
- static `d_open_t` `ngdopen`
- static `d_read_t` `ngdread`
- static `d_write_t` `ngdwrite`
- static `d_poll_t` `ngdpoll`
- static struct `cdevsw` `ngd_cdevsw`

7.88.1 Define Documentation

7.88.1.1 #define DBG do {} while (0)

Definition at line 37 of file `ng_device.c`.

Referenced by `ng_device_constructor()`, `ng_device_disconnect()`, `ng_device_newhook()`, `ng_device_rcvdata()`, `ngdclose()`, `ngdopen()`, `ngdread()`, and `ngdwrite()`.

7.88.1.2 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 63 of file `ng_device.c`.

7.88.1.3 #define MAX_NGD 999

Definition at line 106 of file `ng_device.c`.

Referenced by `ng_device_mod_event()`.

7.88.1.4 #define NGDF_OPEN 0x0001

Definition at line 97 of file ng_device.c.

Referenced by ngdclose(), and ngdopen().

7.88.1.5 #define NGDF_RWAIT 0x0002

Definition at line 98 of file ng_device.c.

Referenced by ng_device_rcvdata(), and ngdread().

7.88.2 Typedef Documentation

7.88.2.1 typedef struct ngd_private* priv_p

Definition at line 100 of file ng_device.c.

7.88.3 Function Documentation

7.88.3.1 NETGRAPH_INIT (device, & ngd_tpestruct)

7.88.3.2 static int ng_device_constructor (node_p node) [static]

Definition at line 160 of file ng_device.c.

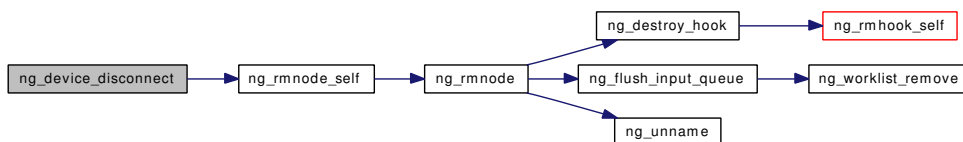
References DBG, ifqmaxlen, NG_DEVICE_DEVNAME, NG_NODE_SET_PRIVATE, ngd_cdevsw, and ngd_unit.

7.88.3.3 static int ng_device_disconnect (hook_p hook) [static]

Definition at line 296 of file ng_device.c.

References DBG, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_rmnode_self(), and ngd_unit.

Here is the call graph for this function:



7.88.3.4 static int ng_device_mod_event (module_t, int, void *) [static]

Definition at line 138 of file ng_device.c.

References MAX_NGD, and ngd_unit.

7.88.3.5 `static int ng_device_newhook (node_p node, hook_p hook, const char * name)`
[static]

Definition at line 243 of file ng_device.c.

References `DBG`, and `NG_NODE_PRIVATE`.

7.88.3.6 `static int ng_device_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 262 of file ng_device.c.

References `DBG`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGDF_RWAIT`, and `NGI_GET_M`.

7.88.3.7 `static int ng_device_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 203 of file ng_device.c.

References `ng_mesg::ng_msghdr::cmd`, `ERROUT`, `ng_mesg::header`, `NG_DEVICE_DEVNAME`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_DEVICE_COOKIE`, `NGM_DEVICE_GET_DEVNAME`, and `ng_mesg::ng_msghdr::typecookie`.

7.88.3.8 `static int ng_device_shutdown (node_p node)` [static]

Definition at line 321 of file ng_device.c.

References `NG_NODE_UNREF`.

7.88.3.9 `static int ngdclose (struct cdev * dev, int flag, int mode, struct thread * td)` [static]

Definition at line 352 of file ng_device.c.

References `DBG`, and `NGDF_OPEN`.

7.88.3.10 `static int ngdopen (struct cdev * dev, int flag, int mode, struct thread * td)` [static]

Definition at line 335 of file ng_device.c.

References `DBG`, and `NGDF_OPEN`.

7.88.3.11 `static int ngdpoll (struct cdev * dev, int events, struct thread * td)` [static]

Definition at line 482 of file ng_device.c.

7.88.3.12 `static int ngdread (struct cdev * dev, struct uio * uio, int flag)` [static]

Definition at line 412 of file ng_device.c.

References `DBG`, and `NGDF_RWAIT`.

7.88.3.13 `static int ngdwrite (struct cdev * dev, struct uio * uio, int flag)` [static]

Definition at line 455 of file `ng_device.c`.

References `DBG`, and `NG_SEND_DATA_ONLY`.

7.88.4 Variable Documentation**7.88.4.1** `ng_constructor_t ng_device_constructor` [static]

Definition at line 67 of file `ng_device.c`.

7.88.4.2 `ng_disconnect_t ng_device_disconnect` [static]

Definition at line 72 of file `ng_device.c`.

7.88.4.3 `ng_newhook_t ng_device_newhook` [static]

Definition at line 70 of file `ng_device.c`.

7.88.4.4 `ng_rcvdata_t ng_device_rcvdata` [static]

Definition at line 71 of file `ng_device.c`.

7.88.4.5 `ng_rcvmsg_t ng_device_rcvmsg` [static]

Definition at line 68 of file `ng_device.c`.

7.88.4.6 `ng_shutdown_t ng_device_shutdown` [static]

Definition at line 69 of file `ng_device.c`.

7.88.4.7 `struct cdevsw ngd_cdevsw` [static]**Initial value:**

```
{
    .d_version =    D_VERSION,
    .d_open =      ngdopen,
    .d_close =     ngdclose,
    .d_read =      ngdread,
    .d_write =     ngdwrite,

    .d_poll =      ngdpoll,
    .d_name =      NG_DEVICE_DEVNAME,
}
```

Definition at line 117 of file `ng_device.c`.

Referenced by `ng_device_constructor()`.

7.88.4.8 struct ng_type ngd_tpestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_DEVICE_NODE_TYPE,
    .mod_event =   ng_device_mod_event,
    .constructor = ng_device_constructor,
    .rcvmsg =      ng_device_rcvmsg,
    .shutdown =   ng_device_shutdown,
    .newhook =    ng_device_newhook,
    .rcvdata =    ng_device_rcvdata,
    .disconnect =  ng_device_disconnect,
}
```

Definition at line 75 of file ng_device.c.

7.88.4.9 struct unrhdr* ngd_unit [static]

Definition at line 103 of file ng_device.c.

Referenced by ng_device_constructor(), ng_device_disconnect(), and ng_device_mod_event().

7.88.4.10 d_close_t ngdclose [static]

Definition at line 108 of file ng_device.c.

7.88.4.11 d_open_t ngdopen [static]

Definition at line 109 of file ng_device.c.

7.88.4.12 d_poll_t ngdpoll [static]

Definition at line 115 of file ng_device.c.

7.88.4.13 d_read_t ngdread [static]

Definition at line 110 of file ng_device.c.

7.88.4.14 d_write_t ngdwrite [static]

Definition at line 111 of file ng_device.c.

7.89 /usr/src/sys/netgraph/ng_device.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_DEVICE_NODE_TYPE](#) "device"
- #define [NGM_DEVICE_COOKIE](#) 1091129178
- #define [NG_DEVICE_DEVNAME](#) "ngd"

Enumerations

- enum { [NGM_DEVICE_GET_DEVNAME](#) }

7.89.1 Define Documentation

7.89.1.1 #define [NG_DEVICE_DEVNAME](#) "ngd"

Definition at line 35 of file `ng_device.h`.

Referenced by `ng_device_constructor()`, and `ng_device_rcvmsg()`.

7.89.1.2 #define [NG_DEVICE_NODE_TYPE](#) "device"

Definition at line 33 of file `ng_device.h`.

7.89.1.3 #define [NGM_DEVICE_COOKIE](#) 1091129178

Definition at line 34 of file `ng_device.h`.

Referenced by `ng_device_rcvmsg()`.

7.89.2 Enumeration Type Documentation

7.89.2.1 anonymous enum

Enumerator:

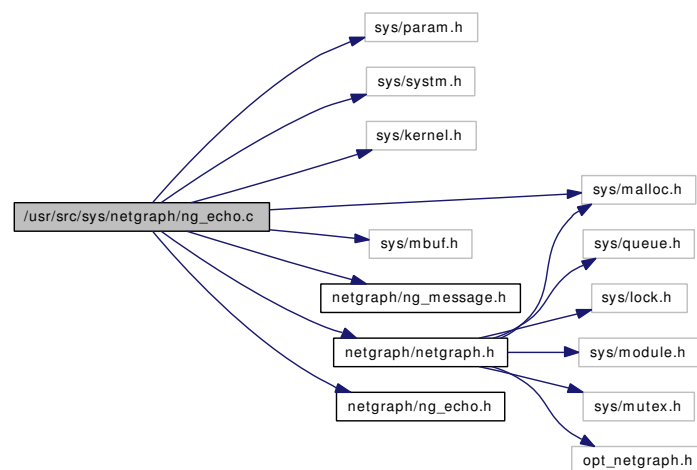
[NGM_DEVICE_GET_DEVNAME](#)

Definition at line 38 of file `ng_device.h`.

7.90 /usr/src/sys/netgraph/ng_echo.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_echo.h>
```

Include dependency graph for ng_echo.c:



Functions

- [NETGRAPH_INIT](#) (echo,&typestruct)
- static int [nge_cons](#) (node_p node)
- static int [nge_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [nge_rcvdata](#) (hook_p hook, item_p item)
- static int [nge_disconnect](#) (hook_p hook)

Variables

- static [ng_constructor_t](#) nge_cons
- static [ng_rcvmsg_t](#) nge_rcvmsg
- static [ng_rcvdata_t](#) nge_rcvdata
- static [ng_disconnect_t](#) nge_disconnect
- static struct [ng_type](#) typestruct

7.90.1 Function Documentation

7.90.1.1 NETGRAPH_INIT (echo, & typestruct)

7.90.1.2 static int nge_cons (node_p node) [static]

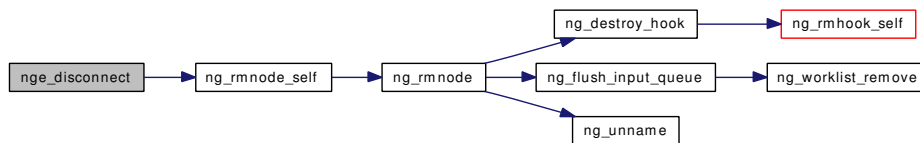
Definition at line 77 of file ng_echo.c.

7.90.1.3 static int nge_disconnect (hook_p hook) [static]

Definition at line 113 of file ng_echo.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, and ng_rmnode_self().

Here is the call graph for this function:



7.90.1.4 static int nge_rcvdata (hook_p hook, item_p item) [static]

Definition at line 101 of file ng_echo.c.

References NG_FWD_ITEM_HOOK.

7.90.1.5 static int nge_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 86 of file ng_echo.c.

References `ng_mesg::ng_msghdr::flags`, `ng_mesg::header`, NG_RESPOND_MSG, NGF_RESP, and NGI_GET_MSG.

7.90.2 Variable Documentation

7.90.2.1 ng_constructor_t nge_cons [static]

Definition at line 60 of file ng_echo.c.

7.90.2.2 ng_disconnect_t nge_disconnect [static]

Definition at line 63 of file ng_echo.c.

7.90.2.3 ng_rcvdata_t nge_rcvdata [static]

Definition at line 62 of file ng_echo.c.

7.90.2.4 `ng_rcvmsg_t nge_rcvmsg` [static]

Definition at line 61 of file ng_echo.c.

7.90.2.5 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_ECHO_NODE_TYPE,
    .constructor = nge_cons,
    .rcvmsg =      nge_rcvmsg,
    .rcvdata =     nge_rcvdata,
    .disconnect =  nge_disconnect,
}
```

Definition at line 66 of file ng_echo.c.

7.91 /usr/src/sys/netgraph/ng_echo.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_ECHO_NODE_TYPE](#) "echo"
- #define [NGM_ECHO_COOKIE](#) 884298942

7.91.1 Define Documentation

7.91.1.1 #define [NG_ECHO_NODE_TYPE](#) "echo"

Definition at line 48 of file ng_echo.h.

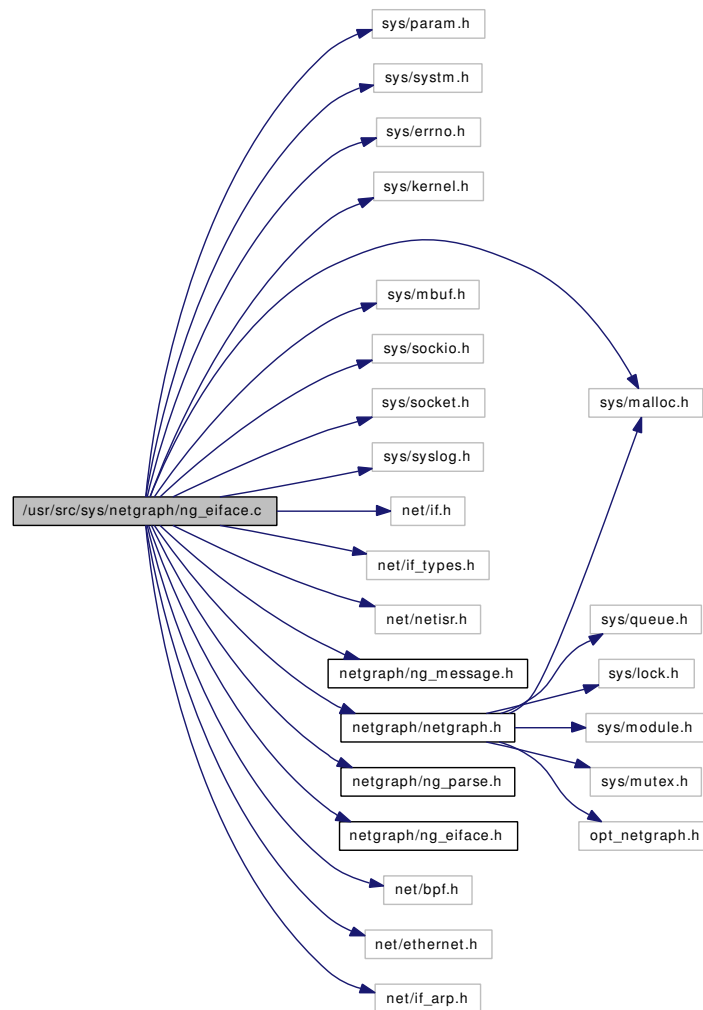
7.91.1.2 #define [NGM_ECHO_COOKIE](#) 884298942

Definition at line 49 of file ng_echo.h.

7.92 /usr/src/sys/netgraph/ng_eiface.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/sockio.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/netisr.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_eiface.h>
#include <net/bpf.h>
#include <net/ethernet.h>
#include <net/if_arp.h>
```

Include dependency graph for ng_eiface.c:



Data Structures

- struct [ng_eiface_private](#)

Defines

- #define [SA_SIZE](#)(s) ((s) → sa_len < sizeof(*(s)) ? sizeof(*(s)) : (s) → sa_len)

Typedefs

- typedef [ng_eiface_private](#) * [priv_p](#)

Functions

- static void [ng_eiface_init](#) (void *xsc)
- static void [ng_eiface_start](#) (struct ifnet *ifp)
- static int [ng_eiface_ioctl](#) (struct ifnet *ifp, u_long cmd, caddr_t data)

- static int `ng_eiface_mod_event` (`module_t`, `int`, `void *`)
- `NETGRAPH_INIT` (`eiface`, `&typestruct`)
- static void `ng_eiface_start2` (`node_p` `node`, `hook_p` `hook`, `void *arg1`, `int arg2`)
- static int `ng_eiface_constructor` (`node_p` `node`)
- static int `ng_eiface_newhook` (`node_p` `node`, `hook_p` `hook`, `const char *name`)
- static int `ng_eiface_rcvmsg` (`node_p` `node`, `item_p` `item`, `hook_p` `lasthook`)
- static int `ng_eiface_rcvdata` (`hook_p` `hook`, `item_p` `item`)
- static int `ng_eiface_rmnode` (`node_p` `node`)
- static int `ng_eiface_disconnect` (`hook_p` `hook`)

Variables

- static struct `ng_cmdlist` `ng_eiface_cmdlist` []
- static `ng_constructor_t` `ng_eiface_constructor`
- static `ng_rcvmsg_t` `ng_eiface_rcvmsg`
- static `ng_shutdown_t` `ng_eiface_rmnode`
- static `ng_newhook_t` `ng_eiface_newhook`
- static `ng_rcvdata_t` `ng_eiface_rcvdata`
- static `ng_disconnect_t` `ng_eiface_disconnect`
- static struct `ng_type` `typestruct`
- static struct `unrhdr *` `ng_eiface_unit`

7.92.1 Define Documentation

7.92.1.1 `#define SA_SIZE(s) ((s) → sa_len < sizeof(*(s))? sizeof(*(s)): (s) → sa_len)`

Referenced by `ng_eiface_rcvmsg()`.

7.92.2 Typedef Documentation

7.92.2.1 `typedef struct ng_eiface_private* priv_p`

Definition at line 80 of file `ng_eiface.c`.

7.92.3 Function Documentation

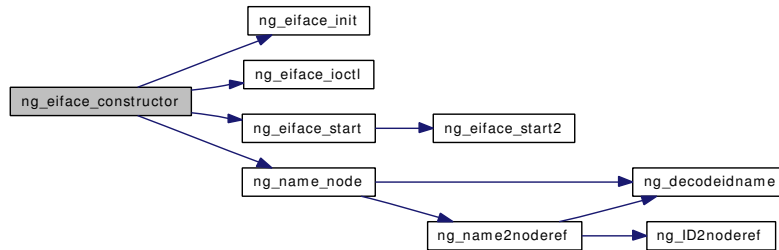
7.92.3.1 `NETGRAPH_INIT` (`eiface`, `& typestruct`)

7.92.3.2 `static int ng_eiface_constructor` (`node_p node`) [static]

Definition at line 326 of file `ng_eiface.c`.

References `NG_EIFACE_EIFACE_NAME`, `ng_eiface_init()`, `ng_eiface_ioctl()`, `ng_eiface_start()`, `ng_eiface_unit`, `ng_name_node()`, and `NG_NODE_SET_PRIVATE`.

Here is the call graph for this function:



7.92.3.3 `static int ng_eiface_disconnect (hook_p hook)` [static]

Definition at line 556 of file `ng_eiface.c`.

References `NG_HOOK_NODE`, and `NG_NODE_PRIVATE`.

7.92.3.4 `static void ng_eiface_init (void *xsc)` [static]

Definition at line 187 of file `ng_eiface.c`.

Referenced by `ng_eiface_constructor()`.

7.92.3.5 `static int ng_eiface_ioctl (struct ifnet *ifp, u_long cmd, caddr_t data)` [static]

Definition at line 124 of file `ng_eiface.c`.

References `NG_EIFACE_MTU_MAX`, and `NG_EIFACE_MTU_MIN`.

Referenced by `ng_eiface_constructor()`.

7.92.3.6 `static int ng_eiface_mod_event (module_t, int, void *)` [static]

Definition at line 568 of file `ng_eiface.c`.

References `ng_eiface_unit`.

7.92.3.7 `static int ng_eiface_newhook (node_p node, hook_p hook, const char *name)` [static]

Definition at line 381 of file `ng_eiface.c`.

References `NG_EIFACE_HOOK_ETHER`, `NG_HOOK_SET_PRIVATE`, and `NG_NODE_PRIVATE`.

7.92.3.8 `static int ng_eiface_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 501 of file `ng_eiface.c`.

References `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, and `NGI_GET_M`.

7.92.3.9 `static int ng_eiface_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 402 of file `ng_eiface.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_EIFACE_COOKIE`, `NGM_EIFACE_GET_IFADDRS`, `NGM_EIFACE_GET_IFNAME`, `NGM_EIFACE_SET`, `NGM_FLOW_COOKIE`, `NGM_LINK_IS_DOWN`, `NGM_LINK_IS_UP`, `SA_SIZE`, and `ng_mesg::ng_msghdr::typecookie`.

7.92.3.10 `static int ng_eiface_rmnode (node_p node)` [static]

Definition at line 538 of file `ng_eiface.c`.

References `ng_eiface_unit`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.92.3.11 `static void ng_eiface_start (struct ifnet * ifp)` [static]

Definition at line 269 of file `ng_eiface.c`.

References `ng_eiface_start2()`, and `ng_send_fn`.

Referenced by `ng_eiface_constructor()`.

Here is the call graph for this function:



7.92.3.12 `static void ng_eiface_start2 (node_p node, hook_p hook, void * arg1, int arg2)` [static]

Definition at line 207 of file `ng_eiface.c`.

References `NG_SEND_DATA_ONLY`.

Referenced by `ng_eiface_start()`.

7.92.4 Variable Documentation

7.92.4.1 `struct ng_cmdlist ng_eiface_cmdlist[]` [static]

Initial value:

```

{
    {
        NGM_EIFACE_COOKIE,
        NGM_EIFACE_GET_IFNAME,
        "getifname",
        NULL,
        &ng_parse_string_type
    },
    {
        NGM_EIFACE_COOKIE,
        NGM_EIFACE_SET,
        "set",
        &ng_parse_enaddr_type,
        NULL
    }
}
  
```

```

    },
    { 0 }
}

```

Definition at line 55 of file ng_eiface.c.

7.92.4.2 [ng_constructor_t ng_eiface_constructor](#) [static]

Definition at line 92 of file ng_eiface.c.

7.92.4.3 [ng_disconnect_t ng_eiface_disconnect](#) [static]

Definition at line 97 of file ng_eiface.c.

7.92.4.4 [ng_newhook_t ng_eiface_newhook](#) [static]

Definition at line 95 of file ng_eiface.c.

7.92.4.5 [ng_rcvdata_t ng_eiface_rcvdata](#) [static]

Definition at line 96 of file ng_eiface.c.

7.92.4.6 [ng_rcvmsg_t ng_eiface_rcvmsg](#) [static]

Definition at line 93 of file ng_eiface.c.

7.92.4.7 [ng_shutdown_t ng_eiface_rmnode](#) [static]

Definition at line 94 of file ng_eiface.c.

7.92.4.8 [struct unrhdr* ng_eiface_unit](#) [static]

Definition at line 114 of file ng_eiface.c.

Referenced by [ng_eiface_constructor\(\)](#), [ng_eiface_mod_event\(\)](#), and [ng_eiface_rmnode\(\)](#).

7.92.4.9 [struct ng_type typestruct](#) [static]

Initial value:

```

{
    .version =      NG_ABI_VERSION,
    .name =         NG_EIFACE_NODE_TYPE,
    .mod_event =    ng_eiface_mod_event,
    .constructor =  ng_eiface_constructor,
    .rcvmsg =       ng_eiface_rcvmsg,
    .shutdown =     ng_eiface_rmnode,
    .newhook =      ng_eiface_newhook,
    .rcvdata =      ng_eiface_rcvdata,
    .disconnect =   ng_eiface_disconnect,
    .cmdlist =      ng_eiface_cmdlist
}

```

Definition at line 100 of file ng_eiface.c.

7.93 /usr/src/sys/netgraph/ng_eiface.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define NG_EIFACE_NODE_TYPE "eiface"`
- `#define NGM_EIFACE_COOKIE 948105892`
- `#define NG_EIFACE_EIFACE_NAME "ngeth"`
- `#define NG_EIFACE_HOOK_ETHER "ether"`
- `#define NG_EIFACE_MTU_MIN 72`
- `#define NG_EIFACE_MTU_MAX 2312`
- `#define NG_EIFACE_MTU_DEFAULT 1500`

Enumerations

- `enum { NGM_EIFACE_GET_IFNAME = 1, NGM_EIFACE_GET_IFADDRS, NGM_EIFACE_SET }`

7.93.1 Define Documentation

7.93.1.1 `#define NG_EIFACE_EIFACE_NAME "ngeth"`

Definition at line 42 of file `ng_eiface.h`.

Referenced by `ng_eiface_constructor()`.

7.93.1.2 `#define NG_EIFACE_HOOK_ETHER "ether"`

Definition at line 45 of file `ng_eiface.h`.

Referenced by `ng_eiface_newhook()`.

7.93.1.3 `#define NG_EIFACE_MTU_DEFAULT 1500`

Definition at line 50 of file `ng_eiface.h`.

7.93.1.4 `#define NG_EIFACE_MTU_MAX 2312`

Definition at line 49 of file `ng_eiface.h`.

Referenced by `ng_eiface_ioctl()`.

7.93.1.5 #define NG_EIFACE_MTU_MIN 72

Definition at line 48 of file ng_iface.h.

Referenced by ng_iface_ioctl().

7.93.1.6 #define NG_EIFACE_NODE_TYPE "eiface"

Definition at line 38 of file ng_iface.h.

7.93.1.7 #define NGM_EIFACE_COOKIE 948105892

Definition at line 39 of file ng_iface.h.

Referenced by ng_iface_rcvmsg().

7.93.2 Enumeration Type Documentation**7.93.2.1 anonymous enum**

Enumerator:

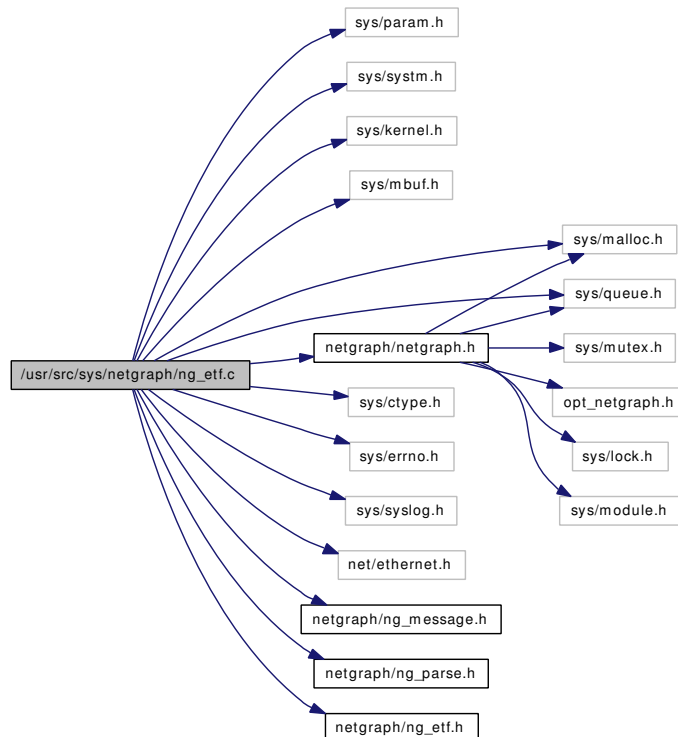
NGM_EIFACE_GET_IFNAME
NGM_EIFACE_GET_IFADDRS
NGM_EIFACE_SET

Definition at line 53 of file ng_iface.h.

7.94 /usr/src/sys/netgraph/ng_etf.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/errno.h>
#include <sys/queue.h>
#include <sys/syslog.h>
#include <net/ethernet.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_etf.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_etf.c:



Data Structures

- struct [ETF_hookinfo](#)

- struct [filter](#)
- struct [ETF](#)

Defines

- #define [M_NETGRAPH ETF](#) M_NETGRAPH
- #define [HASHSIZE](#) 16
- #define [HASH\(et\)](#) (((et)>>12)+((et)>>8)+((et)>>4)+(et)) & 0x0f)

Typedefs

- typedef [ETF](#) * [etf_p](#)

Functions

- [NETGRAPH_INIT](#) (etf,&typestruct)
- [LIST_HEAD](#) (filterhead, filter)
- static struct [filter](#) * [ng_etf_findextry](#) ([etf_p](#) etfp, u_int16_t ethertype)
- static int [ng_etf_constructor](#) ([node_p](#) node)
- static int [ng_etf_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_etf_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_etf_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_etf_shutdown](#) ([node_p](#) node)
- static int [ng_etf_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_etf_constructor](#)
- static [ng_rcvmsg_t](#) [ng_etf_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_etf_shutdown](#)
- static [ng_newhook_t](#) [ng_etf_newhook](#)
- static [ng_rcvdata_t](#) [ng_etf_rcvdata](#)
- static [ng_disconnect_t](#) [ng_etf_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_etf_stat_type_fields](#) [] = NG ETF_STATS_TYPE_INFO
- static struct [ng_parse_type](#) [ng_etf_stat_type](#)
- static struct [ng_parse_struct_field](#) [ng_etf_filter_type_fields](#) [] = NG ETF_FILTER_TYPE_INFO
- static struct [ng_parse_type](#) [ng_etf_filter_type](#)
- static struct [ng_cmdlist](#) [ng_etf_cmdlist](#) []
- static struct [ng_type](#) typestruct

7.94.1 Define Documentation

7.94.1.1 #define [HASH\(et\)](#) (((et)>>12)+((et)>>8)+((et)>>4)+(et)) & 0x0f)

Definition at line 140 of file ng_etf.c.

7.94.1.2 #define HASHSIZE 16

Definition at line 139 of file ng_etf.c.

Referenced by ng_etf_constructor(), ng_etf_disconnect(), ng_vlan_constructor(), and ng_vlan_rcvmsg().

7.94.1.3 #define M_NETGRAPH ETF M_NETGRAPH

Definition at line 58 of file ng_etf.c.

Referenced by ng_etf_constructor(), ng_etf_disconnect(), ng_etf_newhook(), ng_etf_rcvmsg(), and ng_etf_shutdown().

7.94.2 Typedef Documentation

7.94.2.1 typedef struct ETF* etf_p

Definition at line 153 of file ng_etf.c.

7.94.3 Function Documentation

7.94.3.1 LIST_HEAD (filterhead, filter)

7.94.3.2 NETGRAPH_INIT (etf, & typestruct)

7.94.3.3 static int ng_etf_constructor (node_p node) [static]

Definition at line 177 of file ng_etf.c.

References HASHSIZE, M_NETGRAPH ETF, and NG_NODE_SET_PRIVATE.

7.94.3.4 static int ng_etf_disconnect (hook_p hook) [static]

Definition at line 453 of file ng_etf.c.

References HASHSIZE, ETF::hashtable, M_NETGRAPH ETF, NG_HOOK_NODE, and NG_NODE_PRIVATE.

7.94.3.5 static struct filter* ng_etf_findentry (etf_p etfp, u_int16_t ethertype) [static]

Definition at line 156 of file ng_etf.c.

References HASH, and ETF::hashtable.

Referenced by ng_etf_rcvdata(), and ng_etf_rcvmsg().

7.94.3.6 static int ng_etf_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 202 of file ng_etf.c.

References ETF::downstream_hook, ETF_hookinfo::hook, M_NETGRAPH ETF, NG ETF_HOOK_DOWNSTREAM, NG ETF_HOOK_NOMATCH, NG_HOOK_SET_PRIVATE, NG_NODE_PRIVATE, ETF::nomatch_hook, ETF::packets_in, and ETF::packets_out.

7.94.3.7 static int ng_ETF_rcvdata ([hook_p hook](#), [item_p item](#)) [static]

Definition at line 367 of file ng_ETF.c.

References ETF::downstream_hook, ETF_hookinfo::hook, ng_ETF_findentry(), NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, NGL_GET_M, ETF::nomatch_hook, ETF::packets_in, and ETF::packets_out.

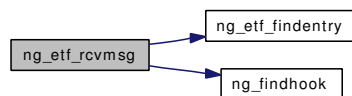
Here is the call graph for this function:

**7.94.3.8** static int ng_ETF_rcvmsg ([node_p node](#), [item_p item](#), [hook_p lasthook](#)) [static]

Definition at line 250 of file ng_ETF.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ETF::downstream_hook, ETF::flags, HASH, ETF::hashtable, ng_mesg::header, ETF_hookinfo::hook, M_NETGRAPH_ETF, ng_ETF_findentry(), ng_findhook(), NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGL_GET_MSG, NGM_ETF_COOKIE, NGM_ETF_GET_STATUS, NGM_ETF_SET_FILTER, NGM_ETF_SET_FLAG, ETF::packets_in, ETF::packets_out, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:

**7.94.3.9** static int ng_ETF_shutdown ([node_p node](#)) [static]

Definition at line 437 of file ng_ETF.c.

References M_NETGRAPH_ETF, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ETF::node.

7.94.4 Variable Documentation**7.94.4.1** struct [ng_cmdlist](#) [ng_ETF_cmdlist](#)[] [static]

Initial value:

```

{
    {
        NGM_ETF_COOKIE,
        NGM_ETF_GET_STATUS,
        "getstatus",
        NULL,
        &ng_ETF_stat_type,
    },
    {

```

```

    NGM ETF_COOKIE,
    NGM ETF_SET_FLAG,
    "setflag",
    &ng_parse_int32_type,
    NULL
},
{
    NGM ETF_COOKIE,
    NGM ETF_SET_FILTER,
    "setfilter",
    &ng_etf_filter_type,
    NULL
},
{ 0 }
}

```

Definition at line 89 of file ng_etf.c.

7.94.4.2 [ng_constructor_t ng_etf_constructor](#) [static]

Definition at line 66 of file ng_etf.c.

7.94.4.3 [ng_disconnect_t ng_etf_disconnect](#) [static]

Definition at line 71 of file ng_etf.c.

7.94.4.4 [struct ng_parse_type ng_etf_filter_type](#) [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_etf_filter_type_fields
}

```

Definition at line 83 of file ng_etf.c.

7.94.4.5 [struct ng_parse_struct_field ng_etf_filter_type_fields\[\] = NG ETF_FILTER_TYPE_ INFO](#) [static]

Definition at line 82 of file ng_etf.c.

7.94.4.6 [ng_newhook_t ng_etf_newhook](#) [static]

Definition at line 69 of file ng_etf.c.

7.94.4.7 [ng_rcvdata_t ng_etf_rcvdata](#) [static]

Definition at line 70 of file ng_etf.c.

7.94.4.8 [ng_rcvmsg_t ng_etf_rcvmsg](#) [static]

Definition at line 67 of file ng_etf.c.

7.94.4.9 `ng_shutdown_t ng_etf_shutdown` [static]

Definition at line 68 of file ng_etf.c.

7.94.4.10 `struct ng_parse_type ng_etf_stat_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_etf_stat_type_fields
}
```

Definition at line 76 of file ng_etf.c.

7.94.4.11 `struct ng_parse_struct_field ng_etf_stat_type_fields[] = NG ETF_STATS_TYPE_INFO`
[static]

Definition at line 75 of file ng_etf.c.

7.94.4.12 `struct ng_type typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG ETF_NODE_TYPE,
    .constructor =  ng_etf_constructor,
    .rcvmsg =       ng_etf_rcvmsg,
    .shutdown =     ng_etf_shutdown,
    .newhook =      ng_etf_newhook,
    .rcvdata =      ng_etf_rcvdata,
    .disconnect =   ng_etf_disconnect,
    .cmdlist =      ng_etf_cmdlist,
}
```

Definition at line 115 of file ng_etf.c.

7.95 /usr/src/sys/netgraph/ng_etf.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_etfstat](#)
- struct [ng_etffilter](#)

Defines

- #define [NG ETF_NODE_TYPE](#) "etf"
- #define [NGM ETF_COOKIE](#) 983084516
- #define [NG ETF_HOOK_DOWNSTREAM](#) "downstream"
- #define [NG ETF_HOOK_NOMATCH](#) "nomatch"
- #define [NG ETF_STATS_TYPE_INFO](#)
- #define [NG ETF_FILTER_TYPE_INFO](#)

Enumerations

- enum { [NGM ETF_SET_FLAG](#) = 1, [NGM ETF_GET_STATUS](#), [NGM ETF_SET_FILTER](#) }

7.95.1 Define Documentation

7.95.1.1 #define NG ETF_FILTER_TYPE_INFO

Value:

```

{
    { "matchhook",          \
      &ng_parse_hookbuf_type }, \
    { "ethertype",        \
      &ng_parse_uint16_type  }, \
    { NULL }
}
  
```

Definition at line 84 of file ng_etf.h.

7.95.1.2 #define NG ETF_HOOK_DOWNSTREAM "downstream"

Definition at line 49 of file ng_etf.h.

Referenced by [ng_etf_newhook\(\)](#).

7.95.1.3 #define NG ETF_HOOK_NOMATCH "nomatch"

Definition at line 50 of file ng_etf.h.

Referenced by [ng_etf_newhook\(\)](#).

7.95.1.4 #define NG ETF_NODE_TYPE "etf"

Definition at line 40 of file ng_etf.h.

7.95.1.5 #define NG ETF_STATS_TYPE_INFO

Value:

```
{
    { "packets_in",      \
      &ng_parse_uint32_type }, \
    { "packets_out",    \
      &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 69 of file ng_etf.h.

7.95.1.6 #define NGM ETF_COOKIE 983084516

Definition at line 46 of file ng_etf.h.

Referenced by ng_etf_rcvmsg().

7.95.2 Enumeration Type Documentation

7.95.2.1 anonymous enum

Enumerator:

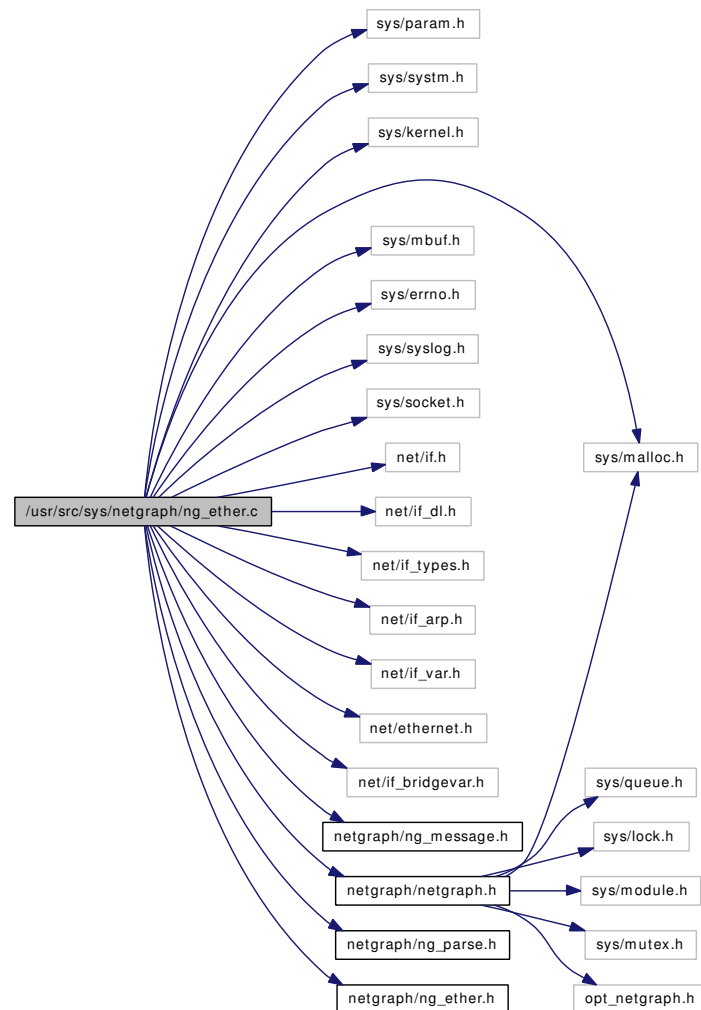
NGM ETF_SET_FLAG
NGM ETF_GET_STATUS
NGM ETF_SET_FILTER

Definition at line 53 of file ng_etf.h.

7.96 /usr/src/sys/netgraph/ng_ether.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/if_dl.h>
#include <net/if_types.h>
#include <net/if_arp.h>
#include <net/if_var.h>
#include <net/ethernet.h>
#include <net/if_bridgevar.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_ether.h>
```

Include dependency graph for ng_ether.c:



Data Structures

- struct [private](#)

Defines

- #define [IFP2NG](#)(ifp) (IFP2AC((ifp)) → ac_netgraph)

Typedefs

- typedef [private](#) * [priv_p](#)

Functions

- static void [ng_ether_input](#) (struct ifnet *ifp, struct mbuf **mp)
- static void [ng_ether_input_orphan](#) (struct ifnet *ifp, struct mbuf *m)
- static int [ng_ether_output](#) (struct ifnet *ifp, struct mbuf **mp)

- static void `ng_ether_attach` (struct ifnet *ifp)
- static void `ng_ether_detach` (struct ifnet *ifp)
- static void `ng_ether_link_state` (struct ifnet *ifp, int state)
- static int `ng_ether_rcv_lower` (node_p node, struct mbuf *m)
- static int `ng_ether_rcv_upper` (node_p node, struct mbuf *m)
- static int `ng_ether_mod_event` (module_t mod, int event, void *data)
- `NETGRAPH_INIT` (ether,&ng_ether_typestruct)
- static int `ng_ether_constructor` (node_p node)
- static int `ng_ether_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_ether_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_ether_rcvdata` (hook_p hook, item_p item)
- static int `ng_ether_shutdown` (node_p node)
- static int `ng_ether_disconnect` (hook_p hook)

Variables

- void(*) `ng_ether_input_p` (struct ifnet *ifp, struct mbuf **mp)
- void(*) `ng_ether_input_orphan_p` (struct ifnet *ifp, struct mbuf *m)
- int(*) `ng_ether_output_p` (struct ifnet *ifp, struct mbuf **mp)
- void(*) `ng_ether_attach_p` (struct ifnet *ifp)
- void(*) `ng_ether_detach_p` (struct ifnet *ifp)
- void(*) `ng_ether_link_state_p` (struct ifnet *ifp, int state)
- static `ng_constructor_t ng_ether_constructor`
- static `ng_rcvmsg_t ng_ether_rcvmsg`
- static `ng_shutdown_t ng_ether_shutdown`
- static `ng_newhook_t ng_ether_newhook`
- static `ng_rcvdata_t ng_ether_rcvdata`
- static `ng_disconnect_t ng_ether_disconnect`
- static struct `ng_cmdlist ng_ether_cmdlist` []
- static struct `ng_type ng_ether_typestruct`

7.96.1 Define Documentation

7.96.1.1 #define IFP2NG(ifp) (IFP2AC((ifp)) → ac_netgraph)

Definition at line 71 of file `ng_ether.c`.

7.96.2 Typedef Documentation

7.96.2.1 typedef struct private* priv_p

Definition at line 84 of file `ng_ether.c`.

7.96.3 Function Documentation

7.96.3.1 NETGRAPH_INIT (ether, & ng_ether_tpestruct)

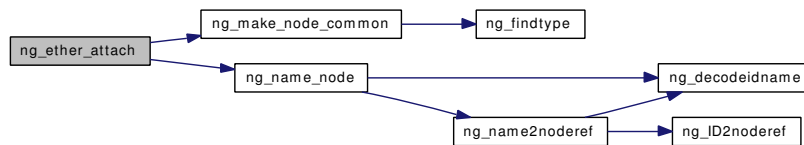
7.96.3.2 static void ng_ether_attach (struct ifnet * ifp) [static]

Definition at line 280 of file ng_ether.c.

References IFP2NG, ng_ether_tpestruct, ng_make_node_common(), ng_name_node(), NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

Referenced by ng_ether_mod_event().

Here is the call graph for this function:



7.96.3.3 static int ng_ether_constructor (node_p node) [static]

Definition at line 371 of file ng_ether.c.

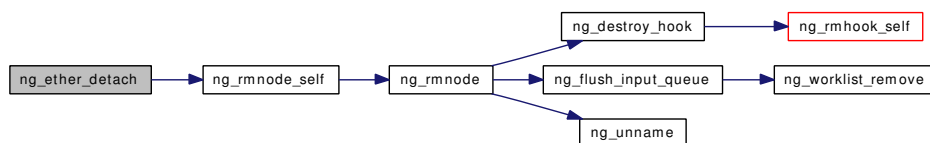
7.96.3.4 static void ng_ether_detach (struct ifnet * ifp) [static]

Definition at line 318 of file ng_ether.c.

References IFP2NG, NG_NODE_PRIVATE, NG_NODE_REALLY_DIE, and ng_rmnode_self().

Referenced by ng_ether_mod_event(), and ng_ether_rcvmsg().

Here is the call graph for this function:

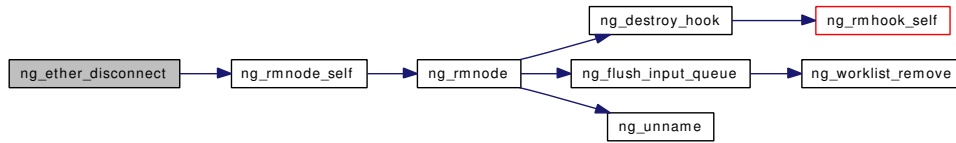


7.96.3.5 static int ng_ether_disconnect (hook_p hook) [static]

Definition at line 688 of file ng_ether.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.96.3.6 static void ng_ether_input (struct ifnet * ifp, struct mbuf ** mp) [static]

Definition at line 222 of file ng_ether.c.

References IFP2NG, NG_NODE_PRIVATE, and NG_SEND_DATA_ONLY.

Referenced by ng_ether_mod_event().

7.96.3.7 static void ng_ether_input_orphan (struct ifnet * ifp, struct mbuf * m) [static]

Definition at line 241 of file ng_ether.c.

References IFP2NG, NG_NODE_PRIVATE, and NG_SEND_DATA_ONLY.

Referenced by ng_ether_mod_event().

7.96.3.8 static void ng_ether_link_state (struct ifnet * ifp, int state) [static]

Definition at line 339 of file ng_ether.c.

References IFP2NG, NG_MKMESSAGE, NG_NODE_PRIVATE, NG_SEND_MSG_HOOK, NGM_FLOW_COOKIE, NGM_LINK_IS_DOWN, and NGM_LINK_IS_UP.

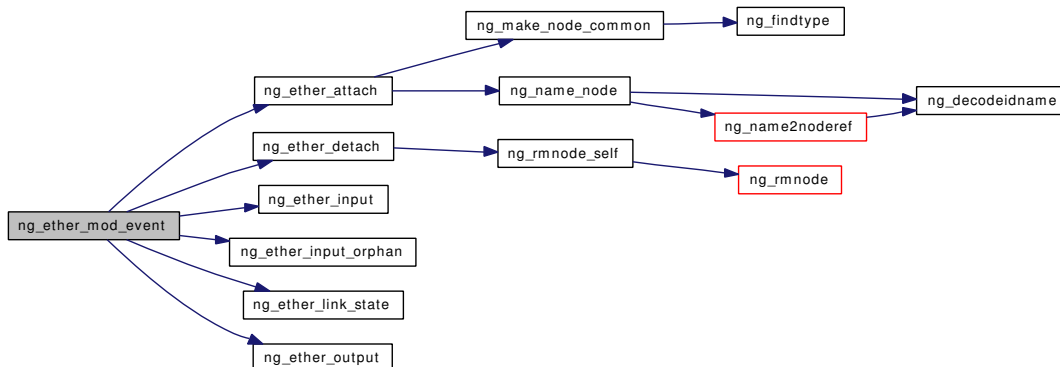
Referenced by ng_ether_mod_event().

7.96.3.9 static int ng_ether_mod_event (module_t mod, int event, void * data) [static]

Definition at line 716 of file ng_ether.c.

References ng_ether_attach(), ng_ether_attach_p, ng_ether_detach(), ng_ether_detach_p, ng_ether_input(), ng_ether_input_orphan(), ng_ether_input_orphan_p, ng_ether_input_p, ng_ether_link_state(), ng_ether_link_state_p, ng_ether_output(), and ng_ether_output_p.

Here is the call graph for this function:



7.96.3.10 `static int ng_ether_newhook (node_p node, hook_p hook, const char * name)`
[static]

Definition at line 380 of file ng_ether.c.

References NG_ETHER_HOOK_DIVERT, NG_ETHER_HOOK_LOWER, NG_ETHER_HOOK_ORPHAN, NG_ETHER_HOOK_UPPER, and NG_NODE_PRIVATE.

7.96.3.11 `static int ng_ether_output (struct ifnet * ifp, struct mbuf ** mp)` [static]

Definition at line 260 of file ng_ether.c.

References IFP2NG, NG_NODE_PRIVATE, and NG_SEND_DATA_ONLY.

Referenced by ng_ether_mod_event().

7.96.3.12 `static int ng_ether_rcv_lower (node_p node, struct mbuf * m)` [static]

Definition at line 582 of file ng_ether.c.

References NG_FREE_M, and NG_NODE_PRIVATE.

Referenced by ng_ether_rcvdata().

7.96.3.13 `static int ng_ether_rcv_upper (node_p node, struct mbuf * m)` [static]

Definition at line 625 of file ng_ether.c.

References NG_FREE_M, and NG_NODE_PRIVATE.

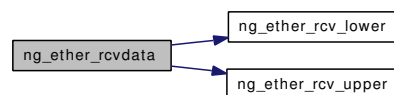
Referenced by ng_ether_rcvdata().

7.96.3.14 `static int ng_ether_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 559 of file ng_ether.c.

References ng_ether_rcv_lower(), ng_ether_rcv_upper(), NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, and NGI_GET_M.

Here is the call graph for this function:



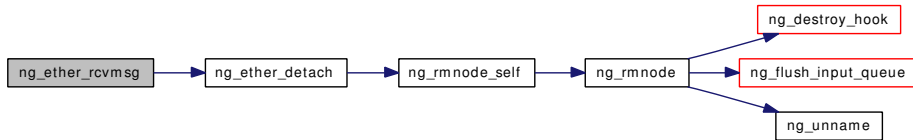
7.96.3.15 `static int ng_ether_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 416 of file ng_ether.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ng_mesg::header, ng_ether_detach(), NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGM_ETHER_ADD_MULTI, NGM_ETHER_COOKIE, NGM_ETHER_DEL_MULTI, NGM_ETHER_DETACH, NGM_ETHER_GET_AUTOSRC, NGM_ETHER_GET_

ENADDR, NGM_ETHER_GET_IFINDEX, NGM_ETHER_GET_IFNAME, NGM_ETHER_GET_PROMISC, NGM_ETHER_SET_AUTOSRC, NGM_ETHER_SET_ENADDR, NGM_ETHER_SET_PROMISC, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.96.3.16 `static int ng_ether_shutdown (node_p node)` [static]

Definition at line 658 of file `ng_ether.c`.

References `ng_node::nd_flags`, `NG_NODE_PRIVATE`, `NG_NODE_REVIVE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `NGF_REALLY_DIE`.

7.96.4 Variable Documentation

7.96.4.1 `void(*) ng_ether_attach_p(struct ifnet *ifp)`

Referenced by `ng_ether_mod_event()`.

7.96.4.2 `struct ng_cmdlist ng_ether_cmdlist[]` [static]

Definition at line 116 of file `ng_ether.c`.

7.96.4.3 `ng_constructor_t ng_ether_constructor` [static]

Definition at line 107 of file `ng_ether.c`.

7.96.4.4 `void(*) ng_ether_detach_p(struct ifnet *ifp)`

Referenced by `ng_ether_mod_event()`.

7.96.4.5 `ng_disconnect_t ng_ether_disconnect` [static]

Definition at line 112 of file `ng_ether.c`.

7.96.4.6 `void(*) ng_ether_input_orphan_p(struct ifnet *ifp, struct mbuf *m)`

Referenced by `ng_ether_mod_event()`.

7.96.4.7 `void(*) ng_ether_input_p(struct ifnet *ifp, struct mbuf **mp)`

Referenced by `ng_ether_mod_event()`.

7.96.4.8 void(*) [ng_ether_link_state_p](#)(struct ifnet *ifp, int state)

Referenced by [ng_ether_mod_event\(\)](#).

7.96.4.9 [ng_newhook_t](#) [ng_ether_newhook](#) [static]

Definition at line 110 of file [ng_ether.c](#).

7.96.4.10 int(*) [ng_ether_output_p](#)(struct ifnet *ifp, struct mbuf **mp)

Referenced by [ng_ether_mod_event\(\)](#).

7.96.4.11 [ng_rcvdata_t](#) [ng_ether_rcvdata](#) [static]

Definition at line 111 of file [ng_ether.c](#).

7.96.4.12 [ng_rcvmsg_t](#) [ng_ether_rcvmsg](#) [static]

Definition at line 108 of file [ng_ether.c](#).

7.96.4.13 [ng_shutdown_t](#) [ng_ether_shutdown](#) [static]

Definition at line 109 of file [ng_ether.c](#).

7.96.4.14 struct [ng_type](#) [ng_ether_typestruct](#) [static]**Initial value:**

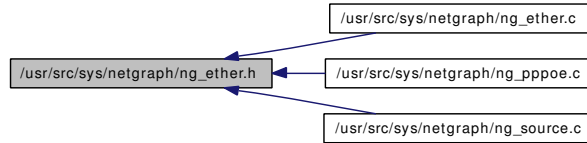
```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_ETHER_NODE_TYPE,
    .mod_event =    ng_ether_mod_event,
    .constructor =  ng_ether_constructor,
    .rcvmsg =       ng_ether_rcvmsg,
    .shutdown =     ng_ether_shutdown,
    .newhook =      ng_ether_newhook,
    .rcvdata =      ng_ether_rcvdata,
    .disconnect =   ng_ether_disconnect,
    .cmdlist =      ng_ether_cmdlist,
}
```

Definition at line 197 of file [ng_ether.c](#).

Referenced by [ng_ether_attach\(\)](#).

7.97 /usr/src/sys/netgraph/ng_ether.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define NG_ETHER_NODE_TYPE "ether"`
- `#define NGM_ETHER_COOKIE 917786906`
- `#define NG_ETHER_HOOK_LOWER "lower"`
- `#define NG_ETHER_HOOK_UPPER "upper"`
- `#define NG_ETHER_HOOK_DIVERT "divert"`
- `#define NG_ETHER_HOOK_ORPHAN "orphans"`

Enumerations

- enum {
 - `NGM_ETHER_GET_IFNAME = 1, NGM_ETHER_GET_IFINDEX, NGM_ETHER_GET_ENADDR, NGM_ETHER_SET_ENADDR,`
 - `NGM_ETHER_GET_PROMISC, NGM_ETHER_SET_PROMISC, NGM_ETHER_GET_AUTOSRC, NGM_ETHER_SET_AUTOSRC,`
 - `NGM_ETHER_ADD_MULTI, NGM_ETHER_DEL_MULTI, NGM_ETHER_DETACH }`

7.97.1 Define Documentation

7.97.1.1 `#define NG_ETHER_HOOK_DIVERT "divert"`

Definition at line 55 of file `ng_ether.h`.

Referenced by `ng_ether_newhook()`.

7.97.1.2 `#define NG_ETHER_HOOK_LOWER "lower"`

Definition at line 53 of file `ng_ether.h`.

Referenced by `ng_ether_newhook()`.

7.97.1.3 `#define NG_ETHER_HOOK_ORPHAN "orphans"`

Definition at line 56 of file `ng_ether.h`.

Referenced by `ng_ether_newhook()`.

7.97.1.4 #define NG_ETHER_HOOK_UPPER "upper"

Definition at line 54 of file ng_ether.h.

Referenced by ng_ether_newhook().

7.97.1.5 #define NG_ETHER_NODE_TYPE "ether"

Definition at line 49 of file ng_ether.h.

7.97.1.6 #define NGM_ETHER_COOKIE 917786906

Definition at line 50 of file ng_ether.h.

Referenced by ng_ether_rcvmsg(), ng_pppoe_connect(), ng_pppoe_rcvmsg(), ng_source_connect(), ng_source_rcvmsg(), and ng_source_set_autosrc().

7.97.2 Enumeration Type Documentation**7.97.2.1 anonymous enum**

Enumerator:

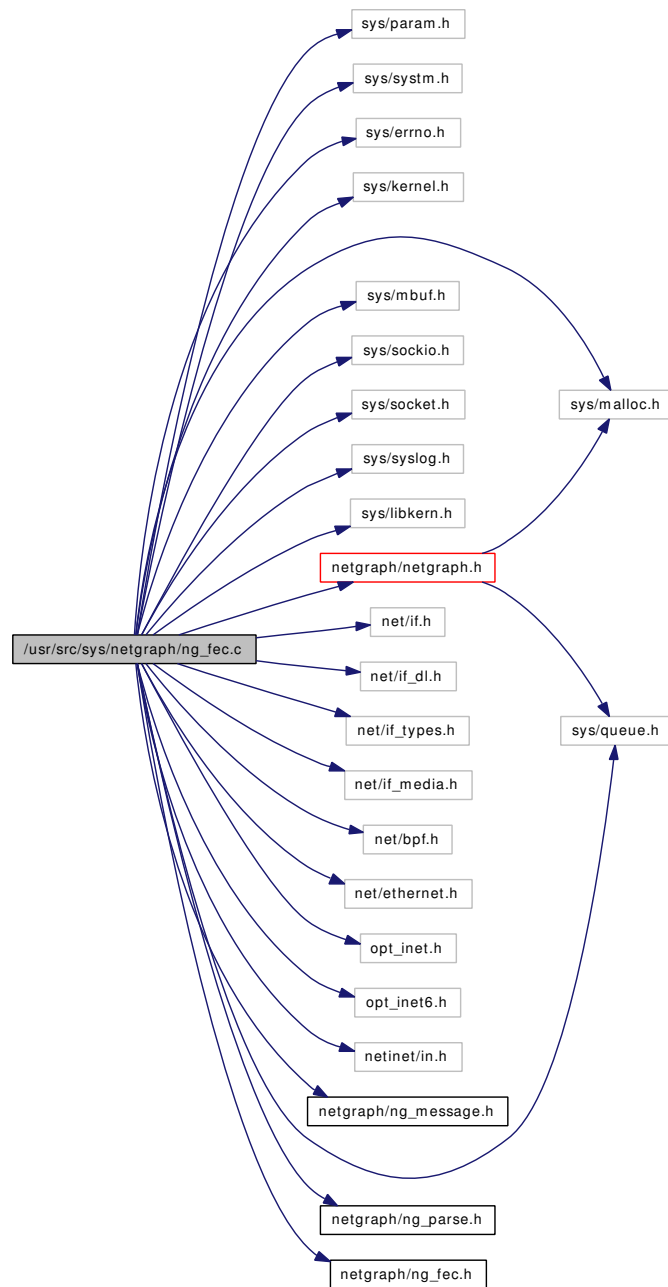
NGM_ETHER_GET_IFNAME
NGM_ETHER_GET_IFINDEX
NGM_ETHER_GET_ENADDR
NGM_ETHER_SET_ENADDR
NGM_ETHER_GET_PROMISC
NGM_ETHER_SET_PROMISC
NGM_ETHER_GET_AUTOSRC
NGM_ETHER_SET_AUTOSRC
NGM_ETHER_ADD_MULTI
NGM_ETHER_DEL_MULTI
NGM_ETHER_DETACH

Definition at line 59 of file ng_ether.h.

7.98 /usr/src/sys/netgraph/ng_fec.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/sockio.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/libkern.h>
#include <sys/queue.h>
#include <net/if.h>
#include <net/if_dl.h>
#include <net/if_types.h>
#include <net/if_media.h>
#include <net/bpf.h>
#include <net/ethernet.h>
#include "opt_inet.h"
#include "opt_inet6.h"
#include <netinet/in.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_fec.h>
```

Include dependency graph for ng_fec.c:



Data Structures

- struct [ng_fec_portlist](#)
- struct [ng_fec_bundle](#)
- struct [ng_fec_private](#)

Defines

- #define [IFP2NG](#)(ifp) ((ifp) → if_afdata[AF_NETGRAPH])
- #define [FEC_BUNDLESIZ](#) 4

- #define [FEC_BTYPE_MAC](#) 0x01
- #define [FEC_BTYPE_INET](#) 0x02
- #define [FEC_BTYPE_INET6](#) 0x03
- #define [UNITS_BITSPERWORD](#) (sizeof(*ng_fec_units) * NBBY)

Typedefs

- typedef [ng_fec_private](#) * [priv_p](#)

Functions

- static void [ng_fec_input](#) (struct ifnet *, struct mbuf *)
- static void [ng_fec_start](#) (struct ifnet *ifp)
- static int [ng_fec_choose_port](#) (struct [ng_fec_bundle](#) *b, struct mbuf *m, struct ifnet **ifp)
- static int [ng_fec_setport](#) (struct ifnet *ifp, u_long [cmd](#), [caddr_t](#) data)
- static void [ng_fec_init](#) (void *arg)
- static void [ng_fec_stop](#) (struct ifnet *ifp)
- static int [ng_fec_ifmedia_upd](#) (struct ifnet *ifp)
- static void [ng_fec_ifmedia_sts](#) (struct ifnet *ifp, struct ifmediareq *ifmr)
- static int [ng_fec_ioctl](#) (struct ifnet *ifp, u_long [cmd](#), [caddr_t](#) data)
- static int [ng_fec_output](#) (struct ifnet *ifp, struct mbuf *m0, struct sockaddr *dst, struct rentry *rt0)
- static void [ng_fec_tick](#) (void *arg)
- static int [ng_fec_addport](#) (struct [ng_fec_private](#) *priv, char *iface)
- static int [ng_fec_delpport](#) (struct [ng_fec_private](#) *priv, char *iface)
- static int [ng_fec_mod_event](#) (module_t, int, void *)
- [NETGRAPH_INIT](#) (fec,&typestruct)
- static __inline int [ng_fec_get_unit](#) (int *unit)
- static __inline void [ng_fec_free_unit](#) (int unit)
- static int [ng_fec_constructor](#) (node_p node)
- static int [ng_fec_rcvmsg](#) (node_p node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_fec_shutdown](#) (node_p node)

Variables

- static [ng_constructor_t](#) [ng_fec_constructor](#)
- static [ng_rcvmsg_t](#) [ng_fec_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_fec_shutdown](#)
- static struct [ng_cmdlist](#) [ng_fec_cmds](#) []
- static struct [ng_type](#) [typestruct](#)
- static int * [ng_fec_units](#) = NULL
- static int [ng_fec_units_len](#) = 0
- static int [ng_units_in_use](#) = 0
- static struct mtx [ng_fec_mtx](#)

7.98.1 Define Documentation

7.98.1.1 #define FEC_BTYPE_INET 0x02

Definition at line 166 of file [ng_fec.c](#).

Referenced by [ng_fec_output\(\)](#), and [ng_fec_rcvmsg\(\)](#).

7.98.1.2 #define FEC_BTYPE_INET6 0x03

Definition at line 167 of file ng_fec.c.

Referenced by ng_fec_rcvmsg().

7.98.1.3 #define FEC_BTYPE_MAC 0x01

Definition at line 165 of file ng_fec.c.

Referenced by ng_fec_addport(), ng_fec_output(), and ng_fec_rcvmsg().

7.98.1.4 #define FEC_BUNDLESIZ 4

Definition at line 143 of file ng_fec.c.

Referenced by ng_fec_addport(), ng_fec_init(), and ng_fec_ioctl().

7.98.1.5 #define IFP2NG(ifp) ((ifp) → if_afdata[AF_NETGRAPH])

Definition at line 137 of file ng_fec.c.

7.98.1.6 #define UNITS_BITSPERWORD (sizeof(*ng_fec_units) * NBBY)

Definition at line 260 of file ng_fec.c.

Referenced by ng_fec_free_unit(), and ng_fec_get_unit().

7.98.2 Typedef Documentation

7.98.2.1 typedef struct ng_fec_private* priv_p

Definition at line 180 of file ng_fec.c.

7.98.3 Function Documentation

7.98.3.1 NETGRAPH_INIT (fec, & typestruct)

7.98.3.2 static int ng_fec_addport (struct ng_fec_private * priv, char * iface) [static]

Definition at line 341 of file ng_fec.c.

References FEC_BTYPE_MAC, ng_fec_private::fec_bundle, FEC_BUNDLESIZ, ng_fec_portlist::fec_if, ng_fec_private::ifp, IFP2NG, ng_fec_input(), ng_fec_private::node, and ng_fec_private::unit.

Referenced by ng_fec_rcvmsg().

Here is the call graph for this function:



7.98.3.3 `static int ng_fec_choose_port (struct ng_fec_bundle * b, struct mbuf * m, struct ifnet ** ifp) [static]`

Definition at line 921 of file `ng_fec.c`.

References `ng_fec_portlist::fec_idx`, `ng_fec_portlist::fec_if`, `ng_fec_portlist::fec_ifstat`, `M_FEC_INET`, `M_FEC_INET6`, and `M_FEC_MAC`.

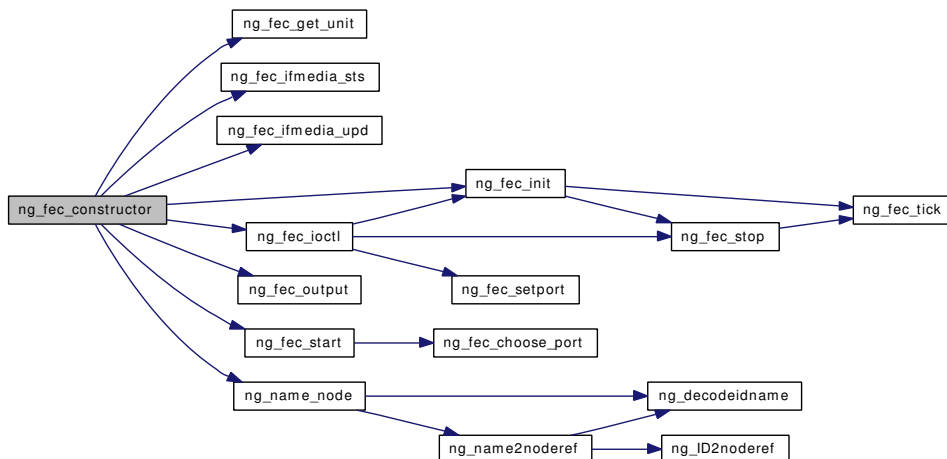
Referenced by `ng_fec_start()`.

7.98.3.4 `static int ng_fec_constructor (node_p node) [static]`

Definition at line 1115 of file `ng_fec.c`.

References `NG_FEC_FEC_NAME`, `NG_FEC_FEC_NAME_MAX`, `ng_fec_get_unit()`, `ng_fec_ifmedia_sts()`, `ng_fec_ifmedia_upd()`, `ng_fec_init()`, `ng_fec_ioctl()`, `NG_FEC_MTU_DEFAULT`, `ng_fec_output()`, `ng_fec_start()`, `ng_name_node()`, and `NG_NODE_SET_PRIVATE`.

Here is the call graph for this function:



7.98.3.5 `static int ng_fec_delport (struct ng_fec_private * priv, char * iface) [static]`

Definition at line 446 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `ng_fec_portlist::fec_if`, `ng_fec_portlist::fec_if_input`, `ng_fec_portlist::fec_mac`, `ng_fec_private::ifp`, `IFP2NG`, and `ng_fec_private::unit`.

Referenced by `ng_fec_rcvmsg()`, and `ng_fec_shutdown()`.

7.98.3.6 `static __inline void ng_fec_free_unit (int unit) [static]`

Definition at line 309 of file `ng_fec.c`.

References `ng_fec_units`, `ng_fec_units_len`, `ng_units_in_use`, and `UNITS_BITSPERWORD`.

Referenced by `ng_fec_shutdown()`.

7.98.3.7 `static __inline int ng_fec_get_unit (int * unit)` [static]

Definition at line 269 of file `ng_fec.c`.

References `ng_fec_units`, `ng_fec_units_len`, `ng_units_in_use`, and `UNITS_BITSPERWORD`.

Referenced by `ng_fec_constructor()`.

7.98.3.8 `static void ng_fec_ifmedia_sts (struct ifnet * ifp, struct ifmediareq * ifmr)` [static]

Definition at line 653 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, and `ng_fec_portlist::fec_ifstat`.

Referenced by `ng_fec_constructor()`.

7.98.3.9 `static int ng_fec_ifmedia_upd (struct ifnet * ifp)` [static]

Definition at line 648 of file `ng_fec.c`.

Referenced by `ng_fec_constructor()`.

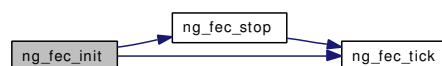
7.98.3.10 `static void ng_fec_init (void * arg)` [static]

Definition at line 536 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `FEC_BUNDLESIZ`, `ng_fec_private::fec_ch`, `ng_fec_portlist::fec_if`, `ng_fec_portlist::fec_ifstat`, `ng_fec_private::ifp`, `ng_fec_stop()`, `ng_fec_tick()`, and `ng_fec_private::unit`.

Referenced by `ng_fec_constructor()`, and `ng_fec_ioctl()`.

Here is the call graph for this function:

**7.98.3.11** `static void ng_fec_input (struct ifnet *, struct mbuf *)` [static]

Definition at line 787 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `ng_fec_portlist::fec_if`, `ng_fec_private::ifp`, `IFP2NG`, and `NG_NODE_PRIVATE`.

Referenced by `ng_fec_addport()`.

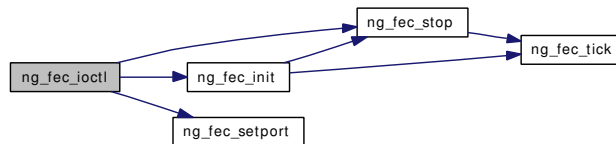
7.98.3.12 `static int ng_fec_ioctl (struct ifnet * ifp, u_long cmd, caddr_t data)` [static]

Definition at line 678 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `FEC_BUNDLESIZ`, `ng_fec_portlist::fec_if`, `ng_fec_private::if_flags`, `ng_fec_private::ifmedia`, `ng_fec_init()`, `NG_FEC_MTU_MAX`, `NG_FEC_MTU_MIN`, `ng_fec_setport()`, `ng_fec_stop()`, and `ng_fec_private::unit`.

Referenced by `ng_fec_constructor()`.

Here is the call graph for this function:



7.98.3.13 `static int ng_fec_mod_event (module_t, int, void *)` [static]

Definition at line 1275 of file `ng_fec.c`.

7.98.3.14 `static int ng_fec_output (struct ifnet * ifp, struct mbuf * m0, struct sockaddr * dst, struct rentry * rt0)` [static]

Definition at line 852 of file `ng_fec.c`.

References `FEC_BTYPE_INET`, `FEC_BTYPE_MAC`, `M_FEC_INET`, `M_FEC_INET6`, and `M_FEC_MAC`.

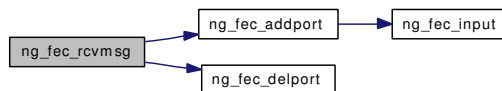
Referenced by `ng_fec_constructor()`.

7.98.3.15 `static int ng_fec_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 1193 of file `ng_fec.c`.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `FEC_BTYPE_INET`, `FEC_BTYPE_INET6`, `FEC_BTYPE_MAC`, `ng_mesg::header`, `ng_fec_addport()`, `ng_fec_delport()`, `NG_FREE_MSG`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_FEC_ADD_IFACE`, `NGM_FEC_COOKIE`, `NGM_FEC_DEL_IFACE`, `NGM_FEC_SET_MODE_INET`, `NGM_FEC_SET_MODE_INET6`, `NGM_FEC_SET_MODE_MAC`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.98.3.16 `static int ng_fec_setport (struct ifnet * ifp, u_long cmd, caddr_t data)` [static]

Definition at line 516 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, and `ng_fec_portlist::fec_if`.

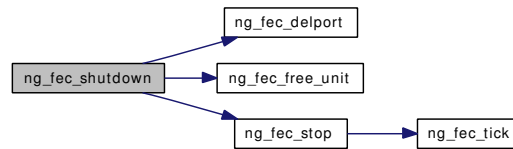
Referenced by `ng_fec_ioctl()`.

7.98.3.17 `static int ng_fec_shutdown (node_p node)` [static]

Definition at line 1247 of file `ng_fec.c`.

References `ng_fec_portlist::fec_if`, `ng_fec_delport()`, `ng_fec_free_unit()`, `ng_fec_stop()`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

Here is the call graph for this function:



7.98.3.18 `static void ng_fec_start (struct ifnet * ifp) [static]`

Definition at line 1040 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `ng_fec_private::if_error`, and `ng_fec_choose_port()`.

Referenced by `ng_fec_constructor()`.

Here is the call graph for this function:



7.98.3.19 `static void ng_fec_stop (struct ifnet * ifp) [static]`

Definition at line 573 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `ng_fec_private::fec_ch`, `ng_fec_portlist::fec_if`, and `ng_fec_tick()`.

Referenced by `ng_fec_init()`, `ng_fec_ioctl()`, and `ng_fec_shutdown()`.

Here is the call graph for this function:



7.98.3.20 `static void ng_fec_tick (void * arg) [static]`

Definition at line 597 of file `ng_fec.c`.

References `ng_fec_private::fec_bundle`, `ng_fec_private::fec_ch`, `ng_fec_portlist::fec_if`, `ng_fec_portlist::fec_ifstat`, `ng_fec_private::ifp`, and `ng_fec_private::unit`.

Referenced by `ng_fec_init()`, and `ng_fec_stop()`.

7.98.4 Variable Documentation

7.98.4.1 `struct cmdlist ng_fec_cmds[] [static]`

Definition at line 210 of file `ng_fec.c`.

7.98.4.2 `ng_constructor_t ng_fec_constructor` [static]

Definition at line 205 of file `ng_fec.c`.

7.98.4.3 `struct mtx ng_fec_mtx` [static]

Definition at line 262 of file `ng_fec.c`.

7.98.4.4 `ng_rcvmsg_t ng_fec_rcvmsg` [static]

Definition at line 206 of file `ng_fec.c`.

7.98.4.5 `ng_shutdown_t ng_fec_shutdown` [static]

Definition at line 207 of file `ng_fec.c`.

7.98.4.6 `int* ng_fec_units = NULL` [static]

Definition at line 256 of file `ng_fec.c`.

Referenced by `ng_fec_free_unit()`, and `ng_fec_get_unit()`.

7.98.4.7 `int ng_fec_units_len = 0` [static]

Definition at line 257 of file `ng_fec.c`.

Referenced by `ng_fec_free_unit()`, and `ng_fec_get_unit()`.

7.98.4.8 `int ng_units_in_use = 0` [static]

Definition at line 258 of file `ng_fec.c`.

Referenced by `ng_fec_free_unit()`, `ng_fec_get_unit()`, `ng_sppp_free_unit()`, and `ng_sppp_get_unit()`.

7.98.4.9 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_FEC_NODE_TYPE,
    .mod_event =   ng_fec_mod_event,
    .constructor = ng_fec_constructor,
    .rcvmsg =      ng_fec_rcvmsg,
    .shutdown =    ng_fec_shutdown,
    .cmdlist =     ng_fec_cmds,
}
```

Definition at line 243 of file `ng_fec.c`.

7.99 /usr/src/sys/netgraph/ng_fec.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_fec_ifname](#)

Defines

- #define [NETISR_FEC](#) 26
- #define [NG_FEC_NODE_TYPE](#) "fec"
- #define [NGM_FEC_COOKIE](#) 983566799
- #define [NG_FEC_FEC_NAME](#) "fec"
- #define [NG_FEC_FEC_NAME_MAX](#) 15
- #define [NG_FEC_MTU_MIN](#) 72
- #define [NG_FEC_MTU_MAX](#) 65535
- #define [NG_FEC_MTU_DEFAULT](#) 1500
- #define [M_FEC_MAC](#) 0x2000
- #define [M_FEC_INET](#) 0x4000
- #define [M_FEC_INET6](#) 0x8000

Enumerations

- enum {
 [NGM_FEC_ADD_IFACE](#), [NGM_FEC_DEL_IFACE](#), [NGM_FEC_SET_MODE_MAC](#), [NGM_FEC_SET_MODE_INET](#),
 [NGM_FEC_SET_MODE_INET6](#) }

7.99.1 Define Documentation

7.99.1.1 #define [M_FEC_INET](#) 0x4000

Definition at line 97 of file [ng_fec.h](#).

Referenced by [ng_fec_choose_port\(\)](#), and [ng_fec_output\(\)](#).

7.99.1.2 #define [M_FEC_INET6](#) 0x8000

Definition at line 98 of file [ng_fec.h](#).

Referenced by [ng_fec_choose_port\(\)](#), and [ng_fec_output\(\)](#).

7.99.1.3 #define M_FEC_MAC 0x2000

Definition at line 96 of file ng_fec.h.

Referenced by ng_fec_choose_port(), and ng_fec_output().

7.99.1.4 #define NETISR_FEC 26

Definition at line 80 of file ng_fec.h.

7.99.1.5 #define NG_FEC_FEC_NAME "fec"

Definition at line 87 of file ng_fec.h.

Referenced by ng_fec_constructor().

7.99.1.6 #define NG_FEC_FEC_NAME_MAX 15

Definition at line 88 of file ng_fec.h.

Referenced by ng_fec_constructor().

7.99.1.7 #define NG_FEC_MTU_DEFAULT 1500

Definition at line 93 of file ng_fec.h.

Referenced by ng_fec_constructor().

7.99.1.8 #define NG_FEC_MTU_MAX 65535

Definition at line 92 of file ng_fec.h.

Referenced by ng_fec_ioctl().

7.99.1.9 #define NG_FEC_MTU_MIN 72

Definition at line 91 of file ng_fec.h.

Referenced by ng_fec_ioctl().

7.99.1.10 #define NG_FEC_NODE_TYPE "fec"

Definition at line 83 of file ng_fec.h.

7.99.1.11 #define NGM_FEC_COOKIE 983566799

Definition at line 84 of file ng_fec.h.

Referenced by ng_fec_rcvmsg().

7.99.2 Enumeration Type Documentation

7.99.2.1 anonymous enum

Enumerator:

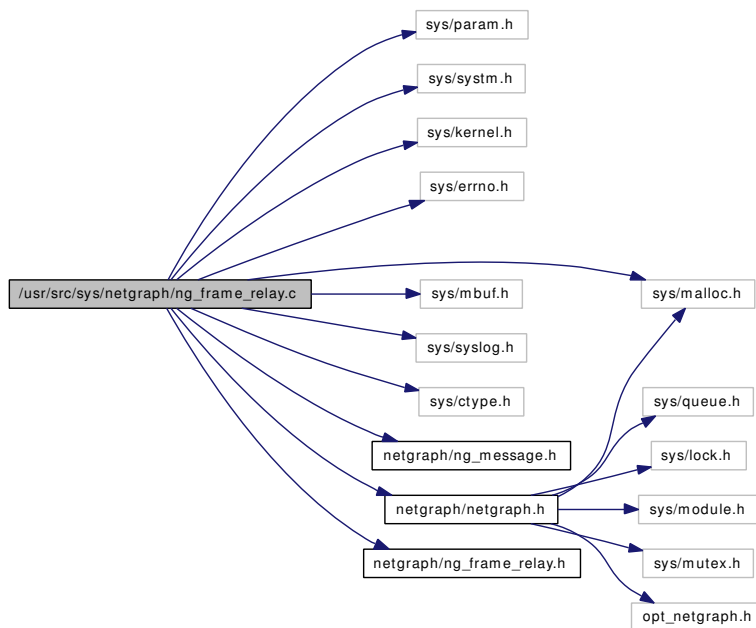
NGM_FEC_ADD_IFACE
NGM_FEC_DEL_IFACE
NGM_FEC_SET_MODE_MAC
NGM_FEC_SET_MODE_INET
NGM_FEC_SET_MODE_INET6

Definition at line 101 of file ng_fec.h.

7.100 /usr/src/sys/netgraph/ng_frame_relay.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/errno.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/syslog.h>
#include <sys/ctype.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_frame_relay.h>
```

Include dependency graph for `ng_frame_relay.c`:



Data Structures

- struct [ctxinfo](#)
- struct [frmrel_softc](#)
- struct [segment](#)

Defines

- #define [CHAN_VALID](#) 0x01
- #define [CHAN_ACTIVE](#) 0x02

- #define [MAX_CT](#) 16
- #define [CTX_VALID](#) 0x8000
- #define [CTX_VALUE](#) (MAX_CT - 1)
- #define [BYTEX_EA](#) 0x01
- #define [BYTE1_C_R](#) 0x02
- #define [BYTE2_FECN](#) 0x08
- #define [BYTE2_BEEN](#) 0x04
- #define [BYTE2_DE](#) 0x02
- #define [LASTBYTE_D_C](#) 0x02
- #define [SHIFTIN](#)(segment, byte, dlc)
- #define [SHIFTOUT](#)(segment, byte, dlc)

Typedefs

- typedef [frmrel_softc](#) * [sc_p](#)

Functions

- static int [ngfrm_decode](#) ([node_p](#) node, [item_p](#) item)
- static int [ngfrm_addrlen](#) (char *hdr)
- static int [ngfrm_allocate_CTX](#) ([sc_p](#) sc, int dlc)
- [NETGRAPH_INIT](#) (framerelay,&typestruct)
- static int [ngfrm_constructor](#) ([node_p](#) node)
- static int [ngfrm_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ngfrm_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ngfrm_shutdown](#) ([node_p](#) node)
- static int [ngfrm_disconnect](#) ([hook_p](#) hook)

Variables

- static const struct [segment](#) [makeup](#) []
- static [ng_constructor_t](#) [ngfrm_constructor](#)
- static [ng_shutdown_t](#) [ngfrm_shutdown](#)
- static [ng_newhook_t](#) [ngfrm_newhook](#)
- static [ng_rcvdata_t](#) [ngfrm_rcvdata](#)
- static [ng_disconnect_t](#) [ngfrm_disconnect](#)
- static struct [ng_type](#) [typestruct](#)

7.100.1 Define Documentation

7.100.1.1 #define [BYTE1_C_R](#) 0x02

Definition at line 96 of file [ng_frame_relay.c](#).

7.100.1.2 #define [BYTE2_BEEN](#) 0x04

Definition at line 98 of file [ng_frame_relay.c](#).

7.100.1.3 #define BYTE2_DE 0x02

Definition at line 99 of file ng_frame_relay.c.

7.100.1.4 #define BYTE2_FECN 0x08

Definition at line 97 of file ng_frame_relay.c.

7.100.1.5 #define BYTEX_EA 0x01

Definition at line 95 of file ng_frame_relay.c.

Referenced by ngfrm_addrln(), and ngfrm_rcvdata().

7.100.1.6 #define CHAN_ACTIVE 0x02

Definition at line 72 of file ng_frame_relay.c.

Referenced by ngfrm_newhook(), and ngfrm_rcvdata().

7.100.1.7 #define CHAN_VALID 0x01

Definition at line 71 of file ng_frame_relay.c.

Referenced by ngfrm_allocate_CTX().

7.100.1.8 #define CTX_VALID 0x8000

Definition at line 88 of file ng_frame_relay.c.

Referenced by ngfrm_allocate_CTX(), and ngfrm_decode().

7.100.1.9 #define CTX_VALUE (MAX_CT - 1)

Definition at line 89 of file ng_frame_relay.c.

Referenced by ngfrm_allocate_CTX(), and ngfrm_decode().

7.100.1.10 #define LASTBYTE_D_C 0x02

Definition at line 100 of file ng_frame_relay.c.

7.100.1.11 #define MAX_CT 16

Definition at line 77 of file ng_frame_relay.c.

Referenced by ngfrm_allocate_CTX().

7.100.1.12 #define SHIFTIN(segment, byte, dlc)**Value:**

```

{
    (dcli) <=& (segment)->width;
    (dcli) |=
        ((byte) & (segment)->mask) >> (segment)->shift;
}

```

Definition at line 114 of file ng_frame_relay.c.

Referenced by ngfrm_decode().

7.100.1.13 #define SHIFTOUT(segment, byte, dlc)**Value:**

```

{
    (byte) |= ((dcli) << (segment)->shift) & (segment)->mask;
    (dcli) >>= (segment)->width;
}

```

Definition at line 121 of file ng_frame_relay.c.

Referenced by ngfrm_rcvdata().

7.100.2 Typedef Documentation**7.100.2.1 typedef struct frmrel_softc* sc_p**

Definition at line 93 of file ng_frame_relay.c.

7.100.3 Function Documentation**7.100.3.1 NETGRAPH_INIT (framerelay, & typestruct)****7.100.3.2 int ngfrm_addrln (char *hdr) [static]**

Definition at line 312 of file ng_frame_relay.c.

References BYTEX_EA.

Referenced by ngfrm_decode().

7.100.3.3 static int ngfrm_allocate_CTX (sc_p sc, int dcli) [static]

Definition at line 156 of file ng_frame_relay.c.

References CHAN_VALID, CTX_VALID, CTX_VALUE, and MAX_CT.

Referenced by ngfrm_newhook().

7.100.3.4 `static int ngfrm_constructor (node_p node)` [static]

Definition at line 211 of file `ng_frame_relay.c`.

References `NG_NODE_SET_PRIVATE`.

7.100.3.5 `static int ngfrm_decode (node_p node, item_p item)` [static]

Definition at line 416 of file `ng_frame_relay.c`.

References `CTX_VALID`, `CTX_VALUE`, `makeup`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_NODE_PRIVATE`, `ngfrm_addrln()`, `NGI_GET_M`, and `SHIFTIN`.

Referenced by `ngfrm_rcvdata()`.

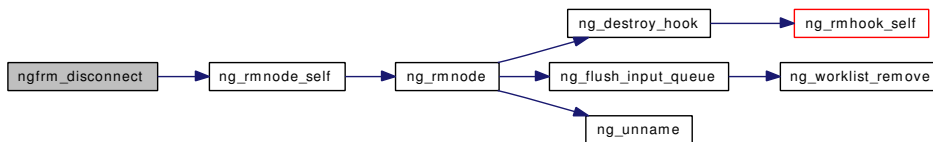
Here is the call graph for this function:

**7.100.3.6** `static int ngfrm_disconnect (hook_p hook)` [static]

Definition at line 496 of file `ng_frame_relay.c`.

References `ctxinfo::dlci`, `ctxinfo::flags`, `ctxinfo::hook`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, and `ng_rmnode_self()`.

Here is the call graph for this function:

**7.100.3.7** `static int ngfrm_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 234 of file `ng_frame_relay.c`.

References `CHAN_ACTIVE`, `ctxinfo::dlci`, `ctxinfo::hook`, `NG_FRAMERELAY_HOOK_DEBUG`, `NG_FRAMERELAY_HOOK_DLCI`, `NG_FRAMERELAY_HOOK_DOWNSTREAM`, `NG_HOOK_SET_PRIVATE`, `NG_NODE_PRIVATE`, and `ngfrm_allocate_CTX()`.

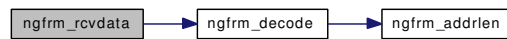
Here is the call graph for this function:

**7.100.3.8** `static int ngfrm_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 329 of file `ng_frame_relay.c`.

References BYTEX_EA, CHAN_ACTIVE, ctxinfo::dlci, ctxinfo::flags, ctxinfo::hook, makeup, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, ngfrm_decode(), NGI_GET_M, and SHIFTOUT.

Here is the call graph for this function:



7.100.3.9 static int ngfrm_shutdown (node_p node) [static]

Definition at line 479 of file ng_frame_relay.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.100.4 Variable Documentation

7.100.4.1 const struct segment makeup[] [static]

Referenced by ngfrm_decode(), and ngfrm_rcvdata().

7.100.4.2 ng_constructor_t ngfrm_constructor [static]

Definition at line 128 of file ng_frame_relay.c.

7.100.4.3 ng_disconnect_t ngfrm_disconnect [static]

Definition at line 132 of file ng_frame_relay.c.

7.100.4.4 ng_newhook_t ngfrm_newhook [static]

Definition at line 130 of file ng_frame_relay.c.

7.100.4.5 ng_rcvdata_t ngfrm_rcvdata [static]

Definition at line 131 of file ng_frame_relay.c.

7.100.4.6 ng_shutdown_t ngfrm_shutdown [static]

Definition at line 129 of file ng_frame_relay.c.

7.100.4.7 struct ng_type typestruct [static]

Initial value:

```

{
    .version =      NG_ABI_VERSION,
    .name =        NG_FRAMERELAY_NODE_TYPE,

```

```
.constructor = ngfrm_constructor,  
.shutdown = ngfrm_shutdown,  
.newhook = ngfrm_newhook,  
.rcvdata = ngfrm_rcvdata,  
.disconnect = ngfrm_disconnect,  
}
```

Definition at line 140 of file ng_frame_relay.c.

7.101 /usr/src/sys/netgraph/ng_frame_relay.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `NG_FRAMERELAY_NODE_TYPE` "frame_relay"
- #define `NGM_FRAMERELAY_COOKIE` 872148478
- #define `NG_FRAMERELAY_HOOK_DEBUG` "debug"
- #define `NG_FRAMERELAY_HOOK_DOWNSTREAM` "downstream"
- #define `NG_FRAMERELAY_HOOK_DLCI` "dlci"

7.101.1 Define Documentation

7.101.1.1 #define `NG_FRAMERELAY_HOOK_DEBUG` "debug"

Definition at line 52 of file `ng_frame_relay.h`.

Referenced by `ngfrm_newhook()`.

7.101.1.2 #define `NG_FRAMERELAY_HOOK_DLCI` "dlci"

Definition at line 54 of file `ng_frame_relay.h`.

Referenced by `ngfrm_newhook()`.

7.101.1.3 #define `NG_FRAMERELAY_HOOK_DOWNSTREAM` "downstream"

Definition at line 53 of file `ng_frame_relay.h`.

Referenced by `ngfrm_newhook()`.

7.101.1.4 #define `NG_FRAMERELAY_NODE_TYPE` "frame_relay"

Definition at line 48 of file `ng_frame_relay.h`.

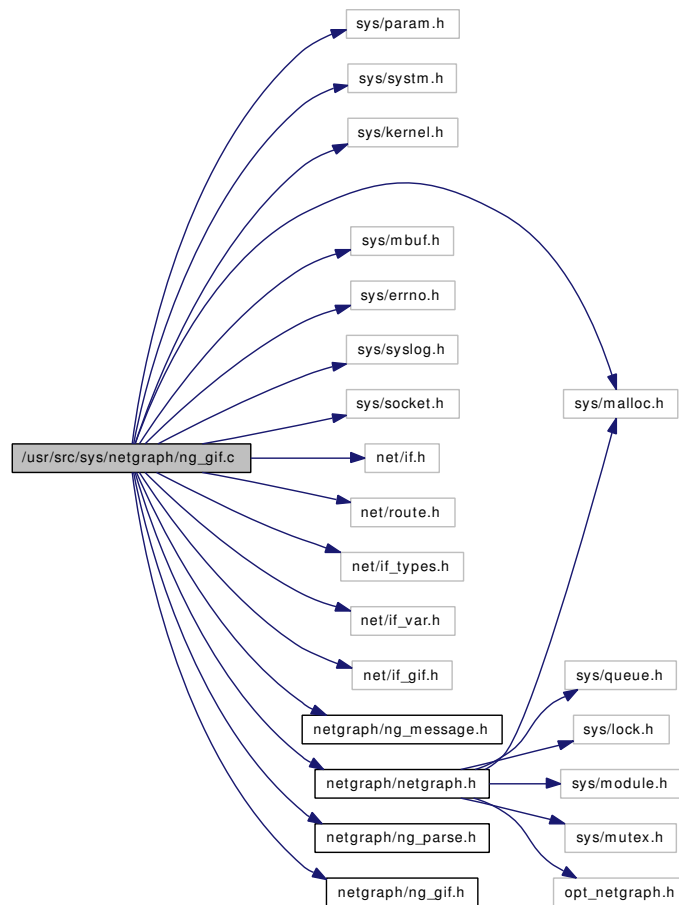
7.101.1.5 #define `NGM_FRAMERELAY_COOKIE` 872148478

Definition at line 49 of file `ng_frame_relay.h`.

7.102 /usr/src/sys/netgraph/ng_gif.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/route.h>
#include <net/if_types.h>
#include <net/if_var.h>
#include <net/if_gif.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_gif.h>
```

Include dependency graph for ng_gif.c:



Data Structures

- struct [private](#)

Defines

- #define [IFP2NG](#)(ifp) ((struct [ng_node](#) *)((struct [gif_softc](#) *) (ifp → if_softc)) → gif_netgraph)
- #define [IFP2NG_SET](#)(ifp, val) (((struct [gif_softc](#) *) (ifp → if_softc)) → gif_netgraph = (val))

Typedefs

- typedef [private](#) * [priv_p](#)

Functions

- static void [ng_gif_input](#) (struct ifnet *ifp, struct mbuf **mp, int af)
- static void [ng_gif_input_orphan](#) (struct ifnet *ifp, struct mbuf *m, int af)
- static void [ng_gif_attach](#) (struct ifnet *ifp)
- static void [ng_gif_detach](#) (struct ifnet *ifp)
- static void [ng_gif_input2](#) ([node_p](#) node, struct mbuf **mp, int af)

- static int `ng_gif_glue_af` (struct mbuf **mp, int af)
- static int `ng_gif_rcv_lower` (node_p node, struct mbuf *m)
- static int `ng_gif_mod_event` (module_t mod, int event, void *data)
- `MODULE_DEPEND` (ng_gif, if_gif, 1, 1, 1)
- `NETGRAPH_INIT` (gif,&ng_gif_typestruct)
- static int `ng_gif_constructor` (node_p node)
- static int `ng_gif_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_gif_connect` (hook_p hook)
- static int `ng_gif_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_gif_rcvdata` (hook_p hook, item_p item)
- static int `ng_gif_shutdown` (node_p node)
- static int `ng_gif_disconnect` (hook_p hook)

Variables

- static `ng_constructor_t ng_gif_constructor`
- static `ng_rcvmsg_t ng_gif_rcvmsg`
- static `ng_shutdown_t ng_gif_shutdown`
- static `ng_newhook_t ng_gif_newhook`
- static `ng_connect_t ng_gif_connect`
- static `ng_rcvdata_t ng_gif_rcvdata`
- static `ng_disconnect_t ng_gif_disconnect`
- static struct `ng_cmdlist ng_gif_cmdlist` []
- static struct `ng_type ng_gif_typestruct`

7.102.1 Define Documentation

7.102.1.1 `#define IFP2NG(ifp) ((struct ng_node *)((struct gif_softc *) (ifp → if_softc)) → gif_netgraph)`

Definition at line 92 of file `ng_gif.c`.

7.102.1.2 `#define IFP2NG_SET(ifp, val) (((struct gif_softc *) (ifp → if_softc)) → gif_netgraph = (val))`

Definition at line 93 of file `ng_gif.c`.

7.102.2 Typedef Documentation

7.102.2.1 `typedef struct private* priv_p`

Definition at line 101 of file `ng_gif.c`.

7.102.3 Function Documentation

7.102.3.1 MODULE_DEPEND (ng_gif, if_gif, 1, 1, 1)

7.102.3.2 NETGRAPH_INIT (gif, & ng_gif_tpestruct)

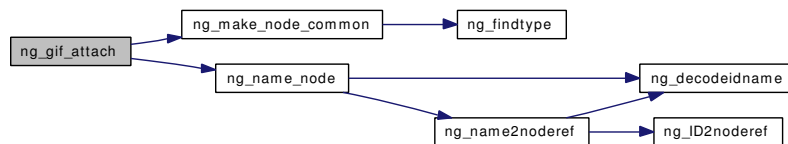
7.102.3.3 static void ng_gif_attach (struct ifnet * ifp) [static]

Definition at line 229 of file ng_gif.c.

References IFP2NG, IFP2NG_SET, ng_gif_tpestruct, ng_make_node_common(), ng_name_node(), NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

Referenced by ng_gif_mod_event().

Here is the call graph for this function:



7.102.3.4 static int ng_gif_connect (hook_p hook) [static]

Definition at line 386 of file ng_gif.c.

References NG_HOOK_FORCE_QUEUE, and NG_HOOK_PEER.

7.102.3.5 static int ng_gif_constructor (node_p node) [static]

Definition at line 341 of file ng_gif.c.

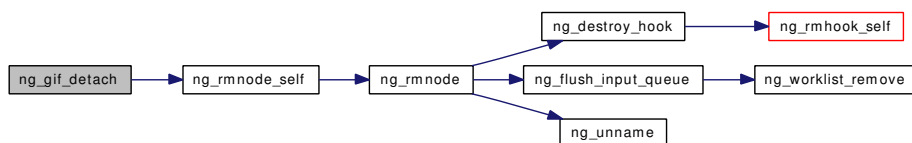
7.102.3.6 static void ng_gif_detach (struct ifnet * ifp) [static]

Definition at line 266 of file ng_gif.c.

References IFP2NG, IFP2NG_SET, NG_NODE_PRIVATE, NG_NODE_REALLY_DIE, and ng_rmnode_self().

Referenced by ng_gif_mod_event().

Here is the call graph for this function:

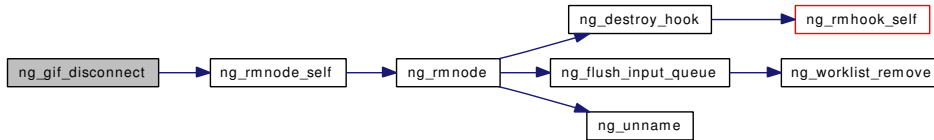


7.102.3.7 static int ng_gif_disconnect (**hook_p** hook) [static]

Definition at line 517 of file ng_gif.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.102.3.8 static int ng_gif_glue_af (struct mbuf ** mp, int af) [static]

Definition at line 290 of file ng_gif.c.

Referenced by ng_gif_input2().

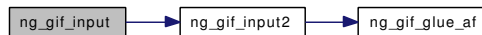
7.102.3.9 static void ng_gif_input (struct ifnet * ifp, struct mbuf ** mp, int af) [static]

Definition at line 170 of file ng_gif.c.

References IFP2NG, ng_gif_input2(), and NG_NODE_PRIVATE.

Referenced by ng_gif_mod_event().

Here is the call graph for this function:



7.102.3.10 static void ng_gif_input2 (node_p node, struct mbuf ** mp, int af) [static]

Definition at line 210 of file ng_gif.c.

References ng_gif_glue_af(), NG_NODE_PRIVATE, and NG_SEND_DATA_ONLY.

Referenced by ng_gif_input(), and ng_gif_input_orphan().

Here is the call graph for this function:



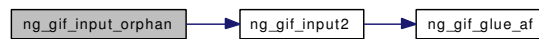
7.102.3.11 static void ng_gif_input_orphan (struct ifnet * ifp, struct mbuf * m, int af) [static]

Definition at line 188 of file ng_gif.c.

References IFP2NG, ng_gif_input2(), and NG_NODE_PRIVATE.

Referenced by `ng_gif_mod_event()`.

Here is the call graph for this function:

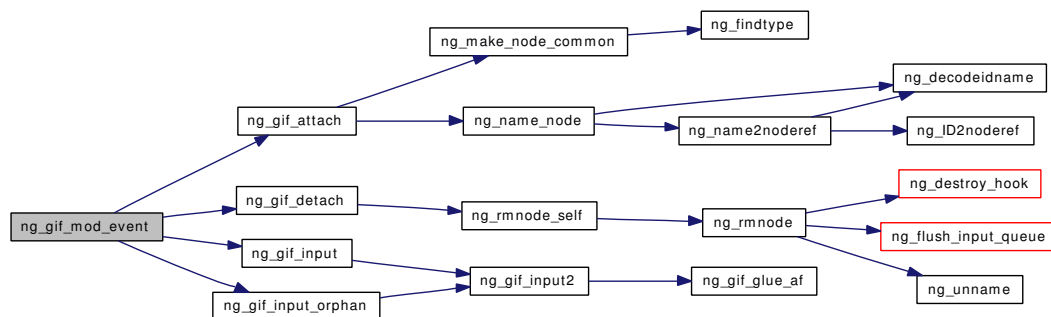


7.102.3.12 `static int ng_gif_mod_event (module_t mod, int event, void * data)` [static]

Definition at line 541 of file `ng_gif.c`.

References `ng_gif_attach()`, `ng_gif_detach()`, `ng_gif_input()`, and `ng_gif_input_orphan()`.

Here is the call graph for this function:



7.102.3.13 `static int ng_gif_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 350 of file `ng_gif.c`.

References `NG_GIF_HOOK_DIVERT`, `NG_GIF_HOOK_LOWER`, `NG_GIF_HOOK_ORPHAN`, and `NG_NODE_PRIVATE`.

7.102.3.14 `static int ng_gif_rcv_lower (node_p node, struct mbuf * m)` [static]

Definition at line 459 of file `ng_gif.c`.

References `NG_FREE_M`, and `NG_NODE_PRIVATE`.

Referenced by `ng_gif_rcvdata()`.

7.102.3.15 `static int ng_gif_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 441 of file `ng_gif.c`.

References `NG_FREE_ITEM`, `ng_gif_rcv_lower()`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, and `NGI_GET_M`.

Here is the call graph for this function:



7.102.3.16 `static int ng_gif_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 396 of file ng_gif.c.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_GIF_COOKIE`, `NGM_GIF_GET_IFINDEX`, `NGM_GIF_GET_IFNAME`, and `ng_mesg::ng_msghdr::typecookie`.

7.102.3.17 `static int ng_gif_shutdown (node_p node)` [static]

Definition at line 493 of file ng_gif.c.

References `ng_node::nd_flags`, `NG_NODE_PRIVATE`, `NG_NODE_REVIVE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `NGF_REALLY_DIE`.

7.102.4 Variable Documentation**7.102.4.1** `struct ng_cmdlist ng_gif_cmdlist[]` [static]

Initial value:

```
{
    {
        NGM_GIF_COOKIE,
        NGM_GIF_GET_IFNAME,
        "getifname",
        NULL,
        &ng_parse_string_type
    },
    {
        NGM_GIF_COOKIE,
        NGM_GIF_GET_IFINDEX,
        "getifindex",
        NULL,
        &ng_parse_int32_type
    },
    { 0 }
}
```

Definition at line 125 of file ng_gif.c.

7.102.4.2 `ng_connect_t ng_gif_connect` [static]

Definition at line 119 of file ng_gif.c.

7.102.4.3 `ng_constructor_t ng_gif_constructor` [static]

Definition at line 115 of file ng_gif.c.

7.102.4.4 `ng_disconnect_t ng_gif_disconnect` [static]

Definition at line 121 of file ng_gif.c.

7.102.4.5 `ng_newhook_t ng_gif_newhook` [static]

Definition at line 118 of file `ng_gif.c`.

7.102.4.6 `ng_rcvdata_t ng_gif_rcvdata` [static]

Definition at line 120 of file `ng_gif.c`.

7.102.4.7 `ng_rcvmsg_t ng_gif_rcvmsg` [static]

Definition at line 116 of file `ng_gif.c`.

7.102.4.8 `ng_shutdown_t ng_gif_shutdown` [static]

Definition at line 117 of file `ng_gif.c`.

7.102.4.9 `struct ng_type ng_gif_typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_GIF_NODE_TYPE,
    .mod_event =    ng_gif_mod_event,
    .constructor =  ng_gif_constructor,
    .rcvmsg =       ng_gif_rcvmsg,
    .shutdown =     ng_gif_shutdown,
    .newhook =      ng_gif_newhook,
    .connect =      ng_gif_connect,
    .rcvdata =      ng_gif_rcvdata,
    .disconnect =   ng_gif_disconnect,
    .cmdlist =      ng_gif_cmdlist,
}
```

Definition at line 143 of file `ng_gif.c`.

Referenced by `ng_gif_attach()`.

7.103 /usr/src/sys/netgraph/ng_gif.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_GIF_NODE_TYPE](#) "gif"
- #define [NGM_GIF_COOKIE](#) 994115727
- #define [NG_GIF_HOOK_LOWER](#) "lower"
- #define [NG_GIF_HOOK_DIVERT](#) "divert"
- #define [NG_GIF_HOOK_ORPHAN](#) "orphans"

Enumerations

- enum { [NGM_GIF_GET_IFNAME](#) = 1, [NGM_GIF_GET_IFINDEX](#) }

7.103.1 Define Documentation

7.103.1.1 #define [NG_GIF_HOOK_DIVERT](#) "divert"

Definition at line 77 of file [ng_gif.h](#).

Referenced by [ng_gif_newhook\(\)](#).

7.103.1.2 #define [NG_GIF_HOOK_LOWER](#) "lower"

Definition at line 76 of file [ng_gif.h](#).

Referenced by [ng_gif_newhook\(\)](#).

7.103.1.3 #define [NG_GIF_HOOK_ORPHAN](#) "orphans"

Definition at line 78 of file [ng_gif.h](#).

Referenced by [ng_gif_newhook\(\)](#).

7.103.1.4 #define [NG_GIF_NODE_TYPE](#) "gif"

Definition at line 72 of file [ng_gif.h](#).

7.103.1.5 #define [NGM_GIF_COOKIE](#) 994115727

Definition at line 73 of file [ng_gif.h](#).

Referenced by [ng_gif_rcvmsg\(\)](#).

7.103.2 Enumeration Type Documentation

7.103.2.1 anonymous enum

Enumerator:

NGM_GIF_GET_IFNAME

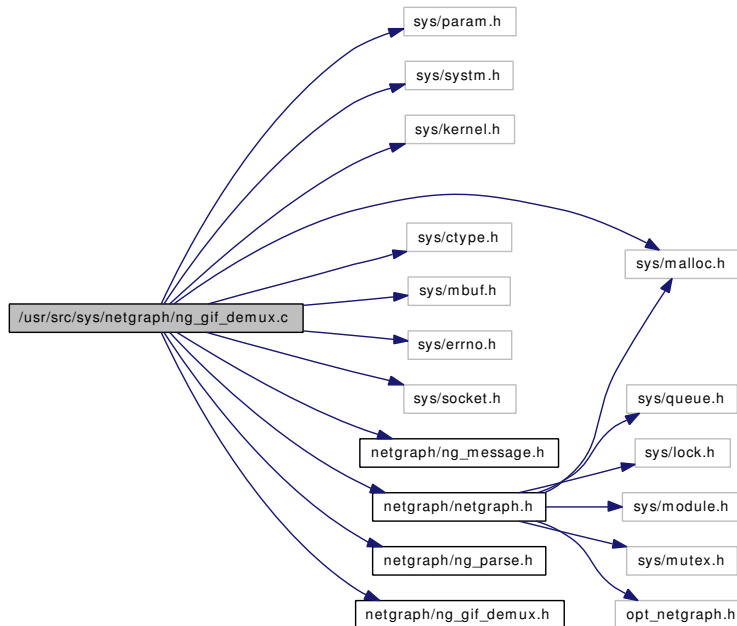
NGM_GIF_GET_IFINDEX

Definition at line 81 of file ng_gif.h.

7.104 /usr/src/sys/netgraph/ng_gif_demux.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/socket.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_gif_demux.h>
```

Include dependency graph for ng_gif_demux.c:



Data Structures

- struct [iffam](#)
- struct [ng_gif_demux_private](#)

Defines

- #define [M_NETGRAPH_GIF_DEMUX](#) M_NETGRAPH

- #define `NUM_FAMILIES` (`sizeof(gFamilies) / sizeof(*gFamilies)`)

Typedefs

- typedef `iffam * iffam_p`
- typedef `ng_gif_demux_private * priv_p`

Functions

- static `iffam_p get_iffam_from_af` (`sa_family_t family`)
- static `iffam_p get_iffam_from_hook` (`priv_p priv`, `hook_p hook`)
- static `iffam_p get_iffam_from_name` (`const char *name`)
- static `hook_p * get_hook_from_iffam` (`priv_p priv`, `iffam_p iffam`)
- `NETGRAPH_INIT` (`gif_demux,&ng_gif_demux_tpestruct`)
- static `int ng_gif_demux_constructor` (`node_p node`)
- static `int ng_gif_demux_newhook` (`node_p node`, `hook_p hook`, `const char *name`)
- static `int ng_gif_demux_rcvmsg` (`node_p node`, `item_p item`, `hook_p lasthook`)
- static `int ng_gif_demux_rcvdata` (`hook_p hook`, `item_p item`)
- static `int ng_gif_demux_shutdown` (`node_p node`)
- static `int ng_gif_demux_disconnect` (`hook_p hook`)

Variables

- static `const struct iffam gFamilies` []
- static `ng_constructor_t ng_gif_demux_constructor`
- static `ng_rcvmsg_t ng_gif_demux_rcvmsg`
- static `ng_shutdown_t ng_gif_demux_shutdown`
- static `ng_newhook_t ng_gif_demux_newhook`
- static `ng_rcvdata_t ng_gif_demux_rcvdata`
- static `ng_disconnect_t ng_gif_demux_disconnect`
- static `struct ng_cmdlist ng_gif_demux_cmdlist` []
- static `struct ng_type ng_gif_demux_tpestruct`

7.104.1 Define Documentation

7.104.1.1 #define M_NETGRAPH_GIF_DEMUX M_NETGRAPH

Definition at line 95 of file `ng_gif_demux.c`.

Referenced by `ng_gif_demux_constructor()`, and `ng_gif_demux_shutdown()`.

7.104.1.2 #define NUM_FAMILIES (sizeof(gFamilies) / sizeof(*gFamilies))

Definition at line 114 of file `ng_gif_demux.c`.

Referenced by `get_iffam_from_af()`, `get_iffam_from_hook()`, and `get_iffam_from_name()`.

7.104.2 Typedef Documentation

7.104.2.1 typedef struct [iffam](#)* [iffam_p](#)

Definition at line 103 of file `ng_gif_demux.c`.

7.104.2.2 typedef struct [ng_gif_demux_private](#)* [priv_p](#)

Definition at line 122 of file `ng_gif_demux.c`.

7.104.3 Function Documentation

7.104.3.1 static `__inline` [hook_p](#) * [get_hook_from_iffam](#) ([priv_p](#) *priv*, [iffam_p](#) *iffam*) [static]

Definition at line 201 of file `ng_gif_demux.c`.

References `gFamilies`.

Referenced by `ng_gif_demux_disconnect()`, `ng_gif_demux_newhook()`, `ng_gif_demux_rcvdata()`, `ng_iface_disconnect()`, `ng_iface_newhook()`, and `ng_iface_send()`.

7.104.3.2 static `__inline` [iffam_p](#) [get_iffam_from_af](#) ([sa_family_t](#) *family*) [static]

Definition at line 169 of file `ng_gif_demux.c`.

References `iffam::family`, `gFamilies`, and `NUM_FAMILIES`.

Referenced by `ng_gif_demux_rcvdata()`, and `ng_iface_send()`.

7.104.3.3 static `__inline` [iffam_p](#) [get_iffam_from_hook](#) ([priv_p](#) *priv*, [hook_p](#) *hook*) [static]

Definition at line 186 of file `ng_gif_demux.c`.

References `gFamilies`, and `NUM_FAMILIES`.

Referenced by `ng_gif_demux_disconnect()`, `ng_gif_demux_rcvdata()`, `ng_iface_disconnect()`, and `ng_iface_rcvdata()`.

7.104.3.4 static `__inline` [iffam_p](#) [get_iffam_from_name](#) (`const char *` *name*) [static]

Definition at line 210 of file `ng_gif_demux.c`.

References `gFamilies`, `iffam::hookname`, and `NUM_FAMILIES`.

Referenced by `ng_gif_demux_newhook()`, and `ng_iface_newhook()`.

7.104.3.5 `NETGRAPH_INIT` ([gif_demux](#), & [ng_gif_demux_tpestruct](#))

7.104.3.6 static int [ng_gif_demux_constructor](#) ([node_p](#) *node*) [static]

Definition at line 231 of file `ng_gif_demux.c`.

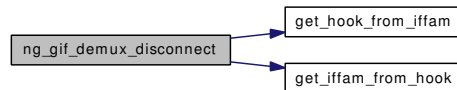
References `M_NETGRAPH_GIF_DEMUX`, and `NG_NODE_SET_PRIVATE`.

7.104.3.7 static int ng_gif_demux_disconnect (hook_p hook) [static]

Definition at line 384 of file ng_gif_demux.c.

References `get_hook_from_iffam()`, `get_iffam_from_hook()`, `NG_HOOK_NODE`, and `NG_NODE_PRIVATE`.

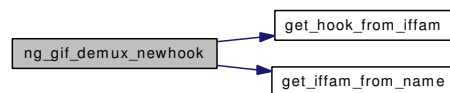
Here is the call graph for this function:

**7.104.3.8 static int ng_gif_demux_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 252 of file ng_gif_demux.c.

References `get_hook_from_iffam()`, `get_iffam_from_name()`, `NG_GIF_DEMUX_HOOK_GIF`, and `NG_NODE_PRIVATE`.

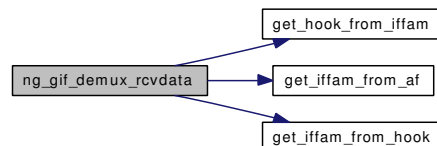
Here is the call graph for this function:

**7.104.3.9 static int ng_gif_demux_rcvdata (hook_p hook, item_p item) [static]**

Definition at line 307 of file ng_gif_demux.c.

References `iffam::family`, `get_hook_from_iffam()`, `get_iffam_from_af()`, `get_iffam_from_hook()`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_ITEM_HOOK`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `NGI_M`.

Here is the call graph for this function:

**7.104.3.10 static int ng_gif_demux_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]**

Definition at line 276 of file ng_gif_demux.c.

References `ng_msgg::ng_msghdr::cmd`, `ng_msgg::header`, `NG_FREE_MSG`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_GIF_DEMUX_COOKIE`, and `ng_msgg::ng_msghdr::typecookie`.

7.104.3.11 `static int ng_gif_demux_shutdown (node_p node)` [static]

Definition at line 370 of file `ng_gif_demux.c`.

References `M_NETGRAPH_GIF_DEMUX`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.104.4 Variable Documentation**7.104.4.1** `const struct iffam gFamilies[]` [static]

Initial value:

```
{
    { AF_INET,      NG_GIF_DEMUX_HOOK_INET  },
    { AF_INET6,    NG_GIF_DEMUX_HOOK_INET6 },
    { AF_APPLETALK, NG_GIF_DEMUX_HOOK_ATALK },
    { AF_IPX,      NG_GIF_DEMUX_HOOK_IPX   },
    { AF_ATM,      NG_GIF_DEMUX_HOOK_ATM   },
    { AF_NATM,     NG_GIF_DEMUX_HOOK_NATM  },
}
```

Definition at line 106 of file `ng_gif_demux.c`.

Referenced by `get_hook_from_iffam()`, `get_iffam_from_af()`, `get_iffam_from_hook()`, and `get_iffam_from_name()`.

7.104.4.2 `struct ng_cmdlist ng_gif_demux_cmdlist[]` [static]

Initial value:

```
{
    { 0 }
}
```

Definition at line 143 of file `ng_gif_demux.c`.

7.104.4.3 `ng_constructor_t ng_gif_demux_constructor` [static]

Definition at line 125 of file `ng_gif_demux.c`.

7.104.4.4 `ng_disconnect_t ng_gif_demux_disconnect` [static]

Definition at line 130 of file `ng_gif_demux.c`.

7.104.4.5 `ng_newhook_t ng_gif_demux_newhook` [static]

Definition at line 128 of file `ng_gif_demux.c`.

7.104.4.6 `ng_rcvdata_t ng_gif_demux_rcvdata` [static]

Definition at line 129 of file `ng_gif_demux.c`.

7.104.4.7 `ng_rcvmsg_t ng_gif_demux_rcvmsg` [static]

Definition at line 126 of file `ng_gif_demux.c`.

7.104.4.8 `ng_shutdown_t ng_gif_demux_shutdown` [static]

Definition at line 127 of file `ng_gif_demux.c`.

7.104.4.9 `struct ng_type ng_gif_demux_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_GIF_DEMUX_NODE_TYPE,
    .constructor = ng_gif_demux_constructor,
    .rcvmsg =      ng_gif_demux_rcvmsg,
    .shutdown =    ng_gif_demux_shutdown,
    .newhook =     ng_gif_demux_newhook,
    .rcvdata =     ng_gif_demux_rcvdata,
    .disconnect =  ng_gif_demux_disconnect,
    .cmdlist =     ng_gif_demux_cmdlist,
}
```

Definition at line 148 of file `ng_gif_demux.c`.

7.105 /usr/src/sys/netgraph/ng_gif_demux.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `NG_GIF_DEMUX_NODE_TYPE` "gif_demux"
- #define `NGM_GIF_DEMUX_COOKIE` 995567329
- #define `NG_GIF_DEMUX_HOOK_GIF` "gif"
- #define `NG_GIF_DEMUX_HOOK_INET` "inet"
- #define `NG_GIF_DEMUX_HOOK_INET6` "inet6"
- #define `NG_GIF_DEMUX_HOOK_ATALK` "atalk"
- #define `NG_GIF_DEMUX_HOOK_IPX` "ipx"
- #define `NG_GIF_DEMUX_HOOK_ATM` "atm"
- #define `NG_GIF_DEMUX_HOOK_NATM` "natm"

7.105.1 Define Documentation

7.105.1.1 #define `NG_GIF_DEMUX_HOOK_ATALK` "atalk"

Definition at line 46 of file `ng_gif_demux.h`.

7.105.1.2 #define `NG_GIF_DEMUX_HOOK_ATM` "atm"

Definition at line 48 of file `ng_gif_demux.h`.

7.105.1.3 #define `NG_GIF_DEMUX_HOOK_GIF` "gif"

Definition at line 43 of file `ng_gif_demux.h`.

Referenced by `ng_gif_demux_newhook()`.

7.105.1.4 #define `NG_GIF_DEMUX_HOOK_INET` "inet"

Definition at line 44 of file `ng_gif_demux.h`.

7.105.1.5 #define `NG_GIF_DEMUX_HOOK_INET6` "inet6"

Definition at line 45 of file `ng_gif_demux.h`.

7.105.1.6 #define `NG_GIF_DEMUX_HOOK_IPX` "ipx"

Definition at line 47 of file `ng_gif_demux.h`.

7.105.1.7 #define NG_GIF_DEMUX_HOOK_NATM "natm"

Definition at line 49 of file ng_gif_demux.h.

7.105.1.8 #define NG_GIF_DEMUX_NODE_TYPE "gif_demux"

Definition at line 39 of file ng_gif_demux.h.

7.105.1.9 #define NGM_GIF_DEMUX_COOKIE 995567329

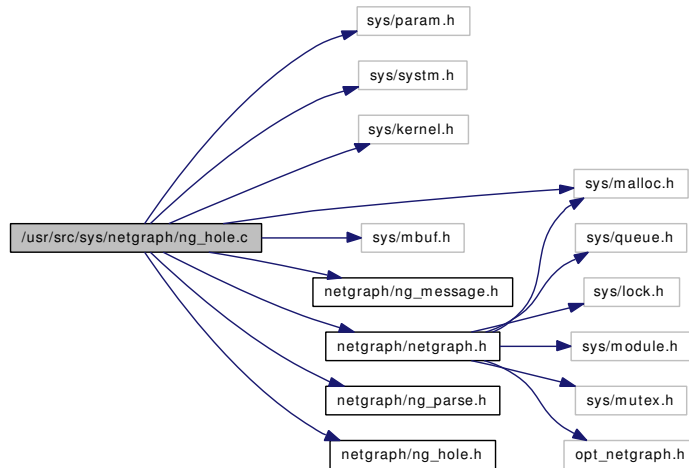
Definition at line 40 of file ng_gif_demux.h.

Referenced by ng_gif_demux_rcvmsg().

7.106 /usr/src/sys/netgraph/ng_hole.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_hole.h>
```

Include dependency graph for ng_hole.c:



Data Structures

- struct [ng_hole_hookinfo](#)

Typedefs

- typedef [ng_hole_hookinfo](#) * [hinfo_p](#)

Functions

- [NETGRAPH_INIT](#) (hole,&typestruct)
- static int [ngh_cons](#) (node_p node)
- static int [ngh_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ngh_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ngh_rcvdata](#) (hook_p hook, item_p item)
- static int [ngh_disconnect](#) (hook_p hook)

Variables

- static struct `ng_parse_struct_field` `ng_hole_hookstat_type_fields` []
- static struct `ng_parse_type` `ng_hole_hookstat_type`
- static struct `ng_cmdlist` `ng_hole_cmdlist` []
- static `ng_constructor_t` `ng_h_cons`
- static `ng_rcvmsg_t` `ng_h_rcvmsg`
- static `ng_newhook_t` `ng_h_newhook`
- static `ng_rcvdata_t` `ng_h_rcvdata`
- static `ng_disconnect_t` `ng_h_disconnect`
- static struct `ng_type` `typestruct`

7.106.1 Typedef Documentation

7.106.1.1 typedef struct `ng_hole_hookinfo*` `hinfo_p`

Definition at line 62 of file `ng_hole.c`.

7.106.2 Function Documentation

7.106.2.1 NETGRAPH_INIT (`hole`, & `typestruct`)

7.106.2.2 static int `ng_h_cons` (`node_p node`) [static]

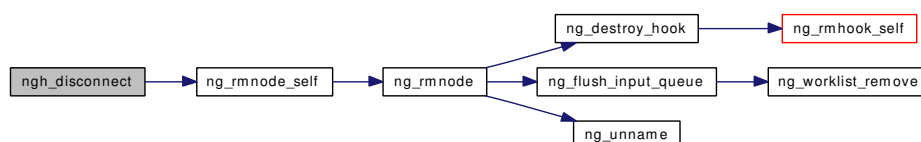
Definition at line 121 of file `ng_hole.c`.

7.106.2.3 static int `ng_h_disconnect` (`hook_p hook`) [static]

Definition at line 219 of file `ng_hole.c`.

References `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_HOOK_SET_PRIVATE`, `NG_NODE_NUMHOOKS`, and `ng_rmnode_self()`.

Here is the call graph for this function:



7.106.2.4 static int `ng_h_newhook` (`node_p node`, `hook_p hook`, `const char * name`) [static]

Definition at line 130 of file `ng_hole.c`.

References `NG_HOOK_SET_PRIVATE`.

7.106.2.5 `static int ngh_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 205 of file ng_hole.c.

References NG_FREE_ITEM, NG_HOOK_PRIVATE, NGI_M, and ng_bpf_hookinfo::stats.

7.106.2.6 `static int ngh_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 146 of file ng_hole.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ng_mesg::header, ng_findhook(), NG_FREE_MSG, NG_HOOK_PRIVATE, NG_HOOKSIZ, NG_MKRESPONSE, NG_RESPOND_MSG, NGI_GET_MSG, NGM_HOLE_CLR_STATS, NGM_HOLE_COOKIE, NGM_HOLE_GET_STATS, NGM_HOLE_GETCLR_STATS, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:

**7.106.3 Variable Documentation****7.106.3.1** `struct ng_cmdlist ng_hole_cmdlist[]` [static]

Initial value:

```

{
    {
        NGM_HOLE_COOKIE,
        NGM_HOLE_GET_STATS,
        "getstats",
        &ng_parse_hookbuf_type,
        &ng_hole_hookstat_type
    },
    {
        NGM_HOLE_COOKIE,
        NGM_HOLE_CLR_STATS,
        "clrstats",
        &ng_parse_hookbuf_type,
        NULL
    },
    {
        NGM_HOLE_COOKIE,
        NGM_HOLE_GETCLR_STATS,
        "getclrstats",
        &ng_parse_hookbuf_type,
        &ng_hole_hookstat_type
    },
    { 0 }
}

```

Definition at line 73 of file ng_hole.c.

7.106.3.2 `struct ng_parse_type ng_hole_hookstat_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_hole_hookstat_type_fields
}
```

Definition at line 67 of file ng_hole.c.

7.106.3.3 struct [ng_parse_struct_field ng_hole_hookstat_type_fields](#)[] [static]

Initial value:

```
NG_HOLE_HOOKSTAT_TYPE_INFO
```

Definition at line 65 of file ng_hole.c.

7.106.3.4 [ng_constructor_t ngh_cons](#) [static]

Definition at line 99 of file ng_hole.c.

7.106.3.5 [ng_disconnect_t ngh_disconnect](#) [static]

Definition at line 103 of file ng_hole.c.

7.106.3.6 [ng_newhook_t ngh_newhook](#) [static]

Definition at line 101 of file ng_hole.c.

7.106.3.7 [ng_rcvdata_t ngh_rcvdata](#) [static]

Definition at line 102 of file ng_hole.c.

7.106.3.8 [ng_rcvmsg_t ngh_rcvmsg](#) [static]

Definition at line 100 of file ng_hole.c.

7.106.3.9 struct [ng_type typestruct](#) [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_HOLE_NODE_TYPE,
    .constructor =  ngh_cons,
    .rcvmsg =      ngh_rcvmsg,
    .newhook =     ngh_newhook,
    .rcvdata =     ngh_rcvdata,
    .disconnect =  ngh_disconnect,
    .cmdlist =     ng_hole_cmdlist,
}
```

Definition at line 105 of file ng_hole.c.

7.107 /usr/src/sys/netgraph/ng_hole.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_hole_hookstat](#)

Defines

- #define [NG_HOLE_NODE_TYPE](#) "hole"
- #define [NGM_HOLE_COOKIE](#) 915433206
- #define [NG_HOLE_HOOKSTAT_TYPE_INFO](#)

Enumerations

- enum { [NGM_HOLE_GET_STATS](#) = 1, [NGM_HOLE_CLR_STATS](#), [NGM_HOLE_GETCLR_STATS](#) }

7.107.1 Define Documentation

7.107.1.1 #define NG_HOLE_HOOKSTAT_TYPE_INFO

Value:

```

{
    { "frames",      &ng_parse_uint64_type },
    { "octets",     &ng_parse_uint64_type },
    { NULL }
}

```

Definition at line 58 of file ng_hole.h.

7.107.1.2 #define NG_HOLE_NODE_TYPE "hole"

Definition at line 48 of file ng_hole.h.

7.107.1.3 #define NGM_HOLE_COOKIE 915433206

Definition at line 49 of file ng_hole.h.

Referenced by ngh_rcvmsg().

7.107.2 Enumeration Type Documentation

7.107.2.1 anonymous enum

Enumerator:

NGM_HOLE_GET_STATS

NGM_HOLE_CLR_STATS

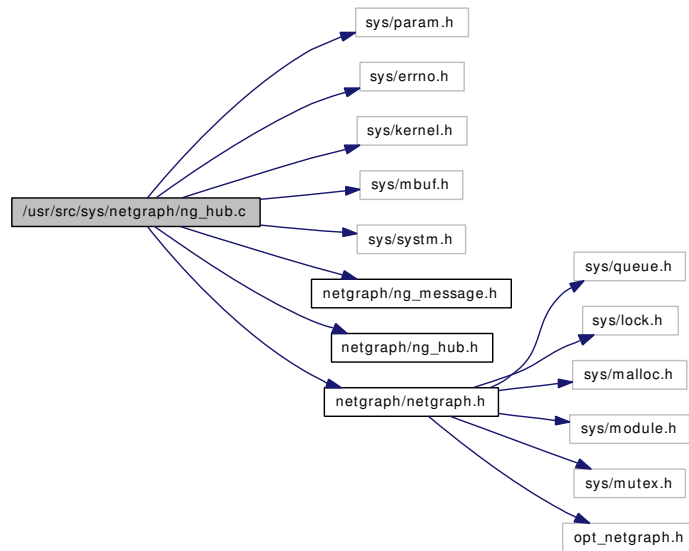
NGM_HOLE_GETCLR_STATS

Definition at line 65 of file ng_hole.h.

7.108 /usr/src/sys/netgraph/ng_hub.c File Reference

```
#include <sys/param.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/system.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_hub.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_hub.c:



Functions

- `NETGRAPH_INIT` (`hub,&ng_hub_tpestruct`)
- static int `ng_hub_constructor` (`node_p node`)
- static int `ng_hub_rcvdata` (`hook_p hook, item_p item`)
- static int `ng_hub_disconnect` (`hook_p hook`)

Variables

- static `ng_constructor_t ng_hub_constructor`
- static `ng_rcvdata_t ng_hub_rcvdata`
- static `ng_disconnect_t ng_hub_disconnect`
- static struct `ng_type ng_hub_tpestruct`

7.108.1 Function Documentation

7.108.1.1 NETGRAPH_INIT (*hub*, & *ng_hub_tpestruct*)

7.108.1.2 static int *ng_hub_constructor* (*node_p node*) [static]

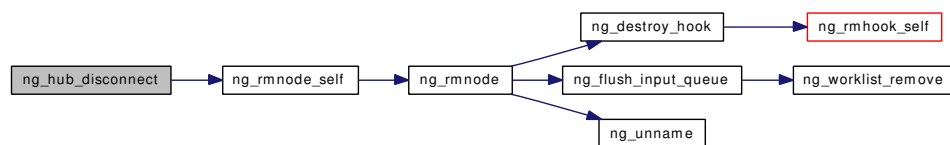
Definition at line 54 of file *ng_hub.c*.

7.108.1.3 static int *ng_hub_disconnect* (*hook_p hook*) [static]

Definition at line 93 of file *ng_hub.c*.

References *NG_HOOK_NODE*, *NG_NODE_IS_VALID*, *NG_NODE_NUMHOOKS*, and *ng_rmnode_self()*.

Here is the call graph for this function:



7.108.1.4 static int *ng_hub_rcvdata* (*hook_p hook*, *item_p item*) [static]

Definition at line 61 of file *ng_hub.c*.

References *NG_FREE_ITEM*, *NG_FWD_ITEM_HOOK*, *NG_HOOK_NODE*, *NG_NODE_NUMHOOKS*, *NG_SEND_DATA_ONLY*, and *NGI_M*.

7.108.2 Variable Documentation

7.108.2.1 *ng_constructor_t ng_hub_constructor* [static]

Definition at line 39 of file *ng_hub.c*.

7.108.2.2 *ng_disconnect_t ng_hub_disconnect* [static]

Definition at line 41 of file *ng_hub.c*.

7.108.2.3 *ng_rcvdata_t ng_hub_rcvdata* [static]

Definition at line 40 of file *ng_hub.c*.

7.108.2.4 struct *ng_type ng_hub_tpestruct* [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_HUB_NODE_TYPE,
    .constructor = ng_hub_constructor,
    .rcvdata =     ng_hub_rcvdata,
    .disconnect =  ng_hub_disconnect,
}
```

Definition at line 43 of file ng_hub.c.

7.109 /usr/src/sys/netgraph/ng_hub.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_HUB_NODE_TYPE](#) "hub"
- #define [NGM_HUB_COOKIE](#) 1082189597

7.109.1 Define Documentation

7.109.1.1 #define [NG_HUB_NODE_TYPE](#) "hub"

Definition at line 33 of file ng_hub.h.

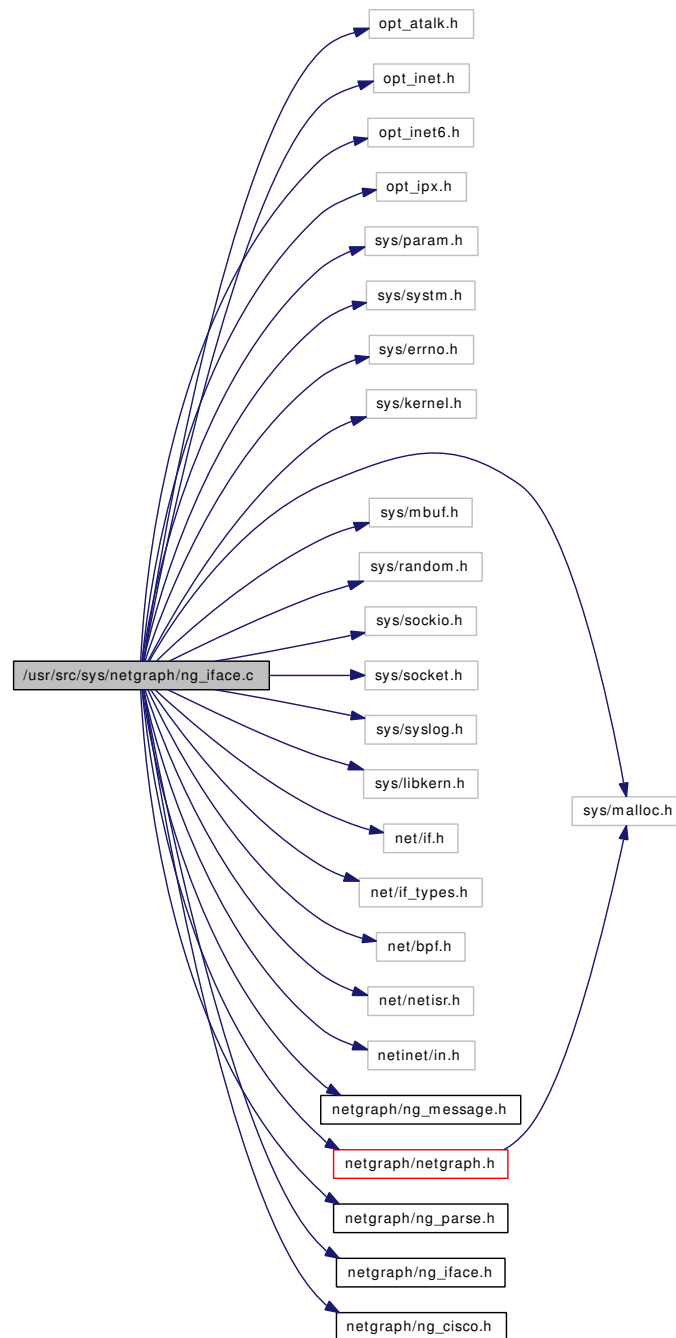
7.109.1.2 #define [NGM_HUB_COOKIE](#) 1082189597

Definition at line 34 of file ng_hub.h.

7.110 /usr/src/sys/netgraph/ng_iface.c File Reference

```
#include "opt_atalk.h"
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipx.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/random.h>
#include <sys/sockio.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/libkern.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/bpf.h>
#include <net/netisr.h>
#include <netinet/in.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_iface.h>
#include <netgraph/ng_cisco.h>
```

Include dependency graph for ng_iface.c:



Data Structures

- struct [iffam](#)
- struct [ng_iface_private](#)

Defines

- #define [M_NETGRAPH_IFACE](#) M_NETGRAPH

- #define `NUM_FAMILIES` (`sizeof(gFamilies) / sizeof(*gFamilies)`)

Typedefs

- typedef `iffam` * `iffam_p`
- typedef `ng_iface_private` * `priv_p`

Functions

- static void `ng_iface_start` (`struct ifnet *ifp`)
- static int `ng_iface_ioctl` (`struct ifnet *ifp`, `u_long cmd`, `caddr_t data`)
- static int `ng_iface_output` (`struct ifnet *ifp`, `struct mbuf *m0`, `struct sockaddr *dst`, `struct rentry *rt0`)
- static void `ng_iface_bpftap` (`struct ifnet *ifp`, `struct mbuf *m`, `sa_family_t family`)
- static int `ng_iface_send` (`struct ifnet *ifp`, `struct mbuf *m`, `sa_family_t sa`)
- static int `ng_iface_mod_event` (`module_t`, `int`, `void *`)
- static `iffam_p` `get_iffam_from_af` (`sa_family_t family`)
- static `iffam_p` `get_iffam_from_hook` (`priv_p priv`, `hook_p hook`)
- static `iffam_p` `get_iffam_from_name` (`const char *name`)
- static `hook_p *` `get_hook_from_iffam` (`priv_p priv`, `iffam_p iffam`)
- `NETGRAPH_INIT` (`iface,&typestruct`)
- static int `ng_iface_constructor` (`node_p node`)
- static int `ng_iface_newhook` (`node_p node`, `hook_p hook`, `const char *name`)
- static int `ng_iface_rcvmsg` (`node_p node`, `item_p item`, `hook_p lasthook`)
- static int `ng_iface_rcvdata` (`hook_p hook`, `item_p item`)
- static int `ng_iface_shutdown` (`node_p node`)
- static int `ng_iface_disconnect` (`hook_p hook`)

Variables

- static const struct `iffam` `gFamilies` []
- static `ng_constructor_t` `ng_iface_constructor`
- static `ng_rcvmsg_t` `ng_iface_rcvmsg`
- static `ng_shutdown_t` `ng_iface_shutdown`
- static `ng_newhook_t` `ng_iface_newhook`
- static `ng_rcvdata_t` `ng_iface_rcvdata`
- static `ng_disconnect_t` `ng_iface_disconnect`
- static struct `ng_parse_struct_field` `ng_cisco_ipaddr_type_fields` [] = `NG_CISCO_IPADDR_TYPE_`
`INFO`
- static struct `ng_parse_type` `ng_cisco_ipaddr_type`
- static struct `ng_cmdlist` `ng_iface_cmds` []
- static struct `ng_type` `typestruct`
- static struct `unrhdr *` `ng_iface_unit`

7.110.1 Define Documentation

7.110.1.1 #define M_NETGRAPH_IFACE M_NETGRAPH

Definition at line 89 of file `ng_iface.c`.

Referenced by `ng_iface_constructor()`, and `ng_iface_shutdown()`.

7.110.1.2 `#define NUM_FAMILIES (sizeof(gFamilies) / sizeof(*gFamilies))`

Definition at line 108 of file `ng_iface.c`.

7.110.2 Typedef Documentation

7.110.2.1 `typedef struct iffam* iffam_p`

Definition at line 97 of file `ng_iface.c`.

7.110.2.2 `typedef struct ng_iface_private* priv_p`

Definition at line 117 of file `ng_iface.c`.

7.110.3 Function Documentation

7.110.3.1 `static __inline hook_p * get_hook_from_iffam (priv_p priv, iffam_p iffam) [static]`

Definition at line 252 of file `ng_iface.c`.

References `gFamilies`.

7.110.3.2 `static __inline iffam_p get_iffam_from_af (sa_family_t family) [static]`

Definition at line 220 of file `ng_iface.c`.

References `iffam::family`, `gFamilies`, and `NUM_FAMILIES`.

7.110.3.3 `static __inline iffam_p get_iffam_from_hook (priv_p priv, hook_p hook) [static]`

Definition at line 237 of file `ng_iface.c`.

References `gFamilies`, and `NUM_FAMILIES`.

7.110.3.4 `static __inline iffam_p get_iffam_from_name (const char * name) [static]`

Definition at line 261 of file `ng_iface.c`.

References `gFamilies`, `iffam::hookname`, and `NUM_FAMILIES`.

7.110.3.5 `NETGRAPH_INIT (iface, & typestruct)`

7.110.3.6 `static void ng_iface_bpftap (struct ifnet * ifp, struct mbuf * m, sa_family_t family) [static]`

Definition at line 418 of file `ng_iface.c`.

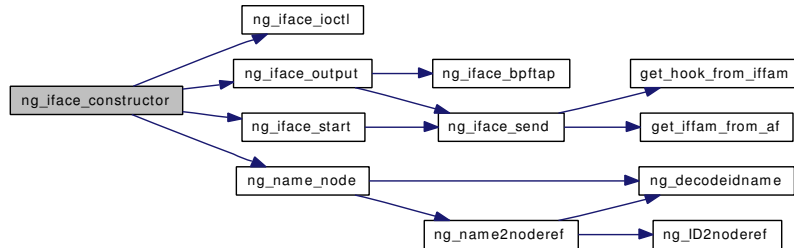
Referenced by `ng_iface_output()`, and `ng_iface_rcvdata()`.

7.110.3.7 static int ng_iface_constructor (node_p node) [static]

Definition at line 506 of file ng_iface.c.

References M_NETGRAPH_IFACE, NG_IFACE_IFACE_NAME, ng_iface_ioctl(), NG_IFACE_MTU_DEFAULT, ng_iface_output(), ng_iface_start(), ng_iface_unit, ng_name_node(), and NG_NODE_SET_PRIVATE.

Here is the call graph for this function:

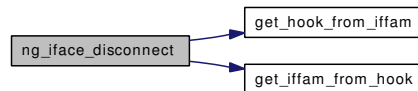


7.110.3.8 static int ng_iface_disconnect (hook_p hook) [static]

Definition at line 785 of file ng_iface.c.

References get_hook_from_iffam(), get_iffam_from_hook(), NG_HOOK_NODE, and NG_NODE_PRIVATE.

Here is the call graph for this function:



7.110.3.9 static int ng_iface_ioctl (struct ifnet * ifp, u_long cmd, caddr_t data) [static]

Definition at line 282 of file ng_iface.c.

References NG_IFACE_MTU_MAX, and NG_IFACE_MTU_MIN.

Referenced by ng_iface_constructor().

7.110.3.10 static int ng_iface_mod_event (module_t, int, void *) [static]

Definition at line 800 of file ng_iface.c.

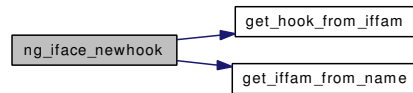
References ng_iface_unit.

7.110.3.11 static int ng_iface_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 565 of file ng_iface.c.

References `get_hook_from_iffam()`, `get_iffam_from_name()`, and `NG_NODE_PRIVATE`.

Here is the call graph for this function:



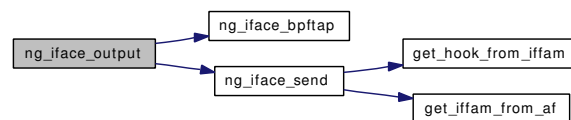
7.110.3.12 `static int ng_iface_output (struct ifnet * ifp, struct mbuf * m0, struct sockaddr * dst, struct rentry * rt0) [static]`

Definition at line 353 of file `ng_iface.c`.

References `ng_iface_bpftap()`, and `ng_iface_send()`.

Referenced by `ng_iface_constructor()`.

Here is the call graph for this function:

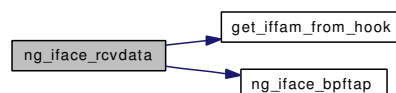


7.110.3.13 `static int ng_iface_rcvdata (hook_p hook, item_p item) [static]`

Definition at line 700 of file `ng_iface.c`.

References `iffam::family`, `get_iffam_from_hook()`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HOOK_NODE`, `ng_iface_bpftap()`, `NG_NODE_PRIVATE`, and `NGI_GET_M`.

Here is the call graph for this function:



7.110.3.14 `static int ng_iface_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]`

Definition at line 583 of file `ng_iface.c`.

References `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_CISCO_COOKIE`, `NGM_CISCO_GET_IPADDR`, `NGM_FLOW_COOKIE`, `NGM_IFACE_BROADCAST`, `NGM_IFACE_COOKIE`, `NGM_IFACE_GET_IFINDEX`, `NGM_IFACE_GET_IFNAME`, `NGM_IFACE_POINT2POINT`, `NGM_LINK_IS_DOWN`, `NGM_LINK_IS_UP`, and `ng_mesg::ng_msghdr::typecookie`.

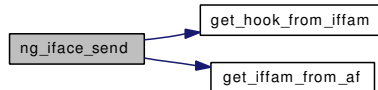
7.110.3.15 `static int ng_iface_send (struct ifnet * ifp, struct mbuf * m, sa_family_t sa)` [static]

Definition at line 433 of file `ng_iface.c`.

References `get_hook_from_iffam()`, `get_iffam_from_af()`, and `NG_SEND_DATA_ONLY`.

Referenced by `ng_iface_output()`, and `ng_iface_start()`.

Here is the call graph for this function:

**7.110.3.16** `static int ng_iface_shutdown (node_p node)` [static]

Definition at line 765 of file `ng_iface.c`.

References `M_NETGRAPH_IFACE`, `ng_iface_unit`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

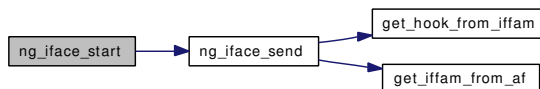
7.110.3.17 `static void ng_iface_start (struct ifnet * ifp)` [static]

Definition at line 396 of file `ng_iface.c`.

References `ng_iface_send()`.

Referenced by `ng_iface_constructor()`.

Here is the call graph for this function:

**7.110.4** Variable Documentation**7.110.4.1** `const struct iffam gFamilies[]` [static]

Initial value:

```

{
    { AF_INET,      NG_IFACE_HOOK_INET      },
    { AF_INET6,    NG_IFACE_HOOK_INET6     },
    { AF_APPLETALK, NG_IFACE_HOOK_ATALK     },
    { AF_IPX,      NG_IFACE_HOOK_IPX       },
    { AF_ATM,      NG_IFACE_HOOK_ATM       },
    { AF_NATM,     NG_IFACE_HOOK_NATM      },
}
  
```

Definition at line 100 of file `ng_iface.c`.

7.110.4.2 struct `ng_parse_type ng_cisco_ipaddr_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_cisco_ipaddr_type_fields
}
```

Definition at line 150 of file `ng_iface.c`.

7.110.4.3 struct `ng_parse_struct_field ng_cisco_ipaddr_type_fields[]` = `NG_CISCO_IPADDR_TYPE_INFO` [static]

Definition at line 149 of file `ng_iface.c`.

7.110.4.4 struct `ng_cmdlist ng_iface_cmds[]` [static]

Definition at line 156 of file `ng_iface.c`.

7.110.4.5 `ng_constructor_t ng_iface_constructor` [static]

Definition at line 134 of file `ng_iface.c`.

7.110.4.6 `ng_disconnect_t ng_iface_disconnect` [static]

Definition at line 139 of file `ng_iface.c`.

7.110.4.7 `ng_newhook_t ng_iface_newhook` [static]

Definition at line 137 of file `ng_iface.c`.

7.110.4.8 `ng_rcvdata_t ng_iface_rcvdata` [static]

Definition at line 138 of file `ng_iface.c`.

7.110.4.9 `ng_rcvmsg_t ng_iface_rcvmsg` [static]

Definition at line 135 of file `ng_iface.c`.

7.110.4.10 `ng_shutdown_t ng_iface_shutdown` [static]

Definition at line 136 of file `ng_iface.c`.

7.110.4.11 struct `unrhdr* ng_iface_unit` [static]

Definition at line 210 of file `ng_iface.c`.

Referenced by `ng_iface_constructor()`, `ng_iface_mod_event()`, and `ng_iface_shutdown()`.

7.110.4.12 struct `ng_type` typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_IFACE_NODE_TYPE,
    .mod_event =    ng_iface_mod_event,
    .constructor = ng_iface_constructor,
    .rcvmsg =       ng_iface_rcvmsg,
    .shutdown =     ng_iface_shutdown,
    .newhook =      ng_iface_newhook,
    .rcvdata =      ng_iface_rcvdata,
    .disconnect =   ng_iface_disconnect,
    .cmdlist =      ng_iface_cmds,
}
```

Definition at line 196 of file `ng_iface.c`.

7.111 /usr/src/sys/netgraph/ng_iface.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_IFACE_NODE_TYPE](#) "iface"
- #define [NGM_IFACE_COOKIE](#) 1108312559
- #define [NG_IFACE_IFACE_NAME](#) "ng"
- #define [NG_IFACE_HOOK_INET](#) "inet"
- #define [NG_IFACE_HOOK_INET6](#) "inet6"
- #define [NG_IFACE_HOOK_ATALK](#) "atalk"
- #define [NG_IFACE_HOOK_IPX](#) "ipx"
- #define [NG_IFACE_HOOK_ATM](#) "atm"
- #define [NG_IFACE_HOOK_NATM](#) "natm"
- #define [NG_IFACE_MTU_MIN](#) 72
- #define [NG_IFACE_MTU_MAX](#) 65535
- #define [NG_IFACE_MTU_DEFAULT](#) 1500

Enumerations

- enum { [NGM_IFACE_GET_IFNAME](#) = 1, [NGM_IFACE_POINT2POINT](#), [NGM_IFACE_BROADCAST](#), [NGM_IFACE_GET_IFINDEX](#) }

7.111.1 Define Documentation

7.111.1.1 #define [NG_IFACE_HOOK_ATALK](#) "atalk"

Definition at line 57 of file ng_iface.h.

7.111.1.2 #define [NG_IFACE_HOOK_ATM](#) "atm"

Definition at line 59 of file ng_iface.h.

7.111.1.3 #define [NG_IFACE_HOOK_INET](#) "inet"

Definition at line 55 of file ng_iface.h.

7.111.1.4 #define [NG_IFACE_HOOK_INET6](#) "inet6"

Definition at line 56 of file ng_iface.h.

7.111.1.5 #define [NG_IFACE_HOOK_IPX](#) "ipx"

Definition at line 58 of file ng_iface.h.

7.111.1.6 #define NG_IFACE_HOOK_NATM "natm"

Definition at line 60 of file ng_iface.h.

7.111.1.7 #define NG_IFACE_IFACE_NAME "ng"

Definition at line 52 of file ng_iface.h.

Referenced by ng_iface_constructor().

7.111.1.8 #define NG_IFACE_MTU_DEFAULT 1500

Definition at line 65 of file ng_iface.h.

Referenced by ng_iface_constructor().

7.111.1.9 #define NG_IFACE_MTU_MAX 65535

Definition at line 64 of file ng_iface.h.

Referenced by ng_iface_ioctl().

7.111.1.10 #define NG_IFACE_MTU_MIN 72

Definition at line 63 of file ng_iface.h.

Referenced by ng_iface_ioctl().

7.111.1.11 #define NG_IFACE_NODE_TYPE "iface"

Definition at line 48 of file ng_iface.h.

7.111.1.12 #define NGM_IFACE_COOKIE 1108312559

Definition at line 49 of file ng_iface.h.

Referenced by ng_iface_rcvmsg().

7.111.2 Enumeration Type Documentation**7.111.2.1 anonymous enum**

Enumerator:

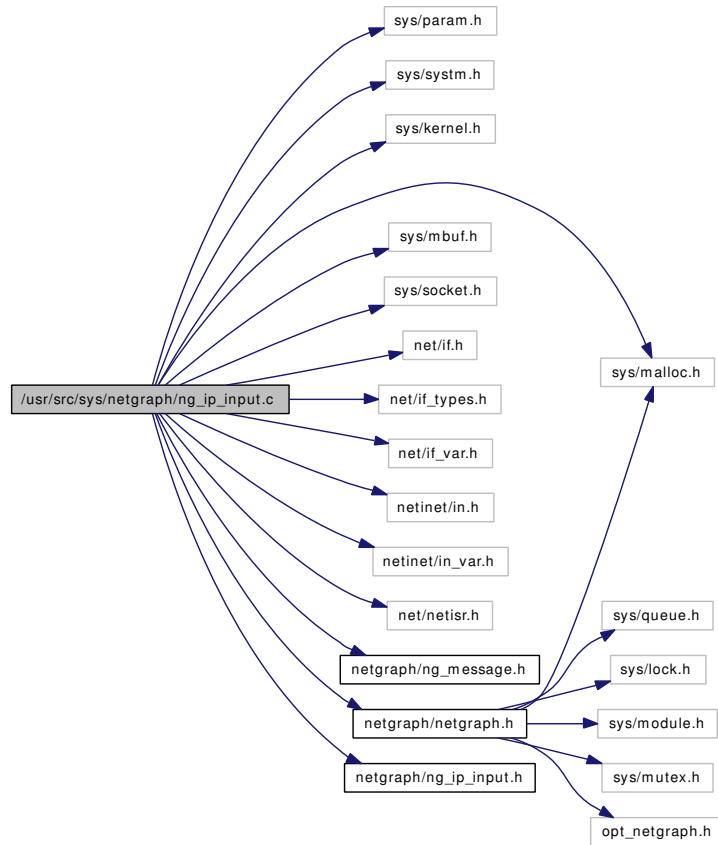
NGM_IFACE_GET_IFNAME
NGM_IFACE_POINT2POINT
NGM_IFACE_BROADCAST
NGM_IFACE_GET_IFINDEX

Definition at line 68 of file ng_iface.h.

7.112 /usr/src/sys/netgraph/ng_ip_input.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/if_var.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <net/netisr.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_ip_input.h>
```

Include dependency graph for ng_ip_input.c:



Functions

- `NETGRAPH_INIT` (`ip_input`, `&typestruct`)
- static int `ngipi_cons` (`node_p` node)
- static int `ngipi_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ngipi_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ngipi_cons`
- static `ng_rcvdata_t` `ngipi_rcvdata`
- static `ng_disconnect_t` `ngipi_disconnect`
- static struct `ng_type` `typestruct`

7.112.1 Function Documentation

7.112.1.1 NETGRAPH_INIT (`ip_input`, & `typestruct`)

7.112.1.2 static int `ngipi_cons` (`node_p` `node`) [static]

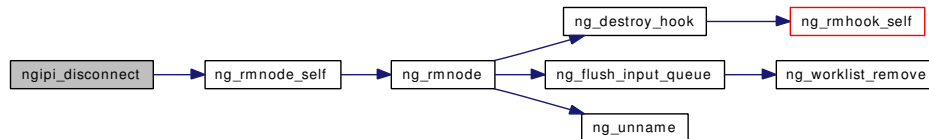
Definition at line 108 of file `ng_ip_input.c`.

7.112.1.3 static int ngipi_disconnect (hook_p hook) [static]

Definition at line 131 of file ng_ip_input.c.

References NG_HOOK_NODE, NG_NODE_NUMHOOKS, and ng_rmnode_self().

Here is the call graph for this function:

**7.112.1.4 static int ngipi_rcvdata (hook_p hook, item_p item) [static]**

Definition at line 117 of file ng_ip_input.c.

References NG_FREE_ITEM, and NGI_GET_M.

7.112.2 Variable Documentation**7.112.2.1 ng_constructor_t ngipi_cons [static]**

Definition at line 91 of file ng_ip_input.c.

7.112.2.2 ng_disconnect_t ngipi_disconnect [static]

Definition at line 93 of file ng_ip_input.c.

7.112.2.3 ng_rcvdata_t ngipi_rcvdata [static]

Definition at line 92 of file ng_ip_input.c.

7.112.2.4 struct ng_type typestruct [static]

Initial value:

```

{
    .version =      NG_ABI_VERSION,
    .name =        NG_IP_INPUT_NODE_TYPE,
    .constructor = ngipi_cons,
    .rcvdata =     ngipi_rcvdata,
    .disconnect =  ngipi_disconnect,
}

```

Definition at line 95 of file ng_ip_input.c.

7.113 /usr/src/sys/netgraph/ng_ip_input.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_IP_INPUT_NODE_TYPE](#) "ip_input"
- #define [NGM_IP_INPUT_COOKIE](#) 994874907

7.113.1 Define Documentation

7.113.1.1 #define [NG_IP_INPUT_NODE_TYPE](#) "ip_input"

Definition at line 74 of file ng_ip_input.h.

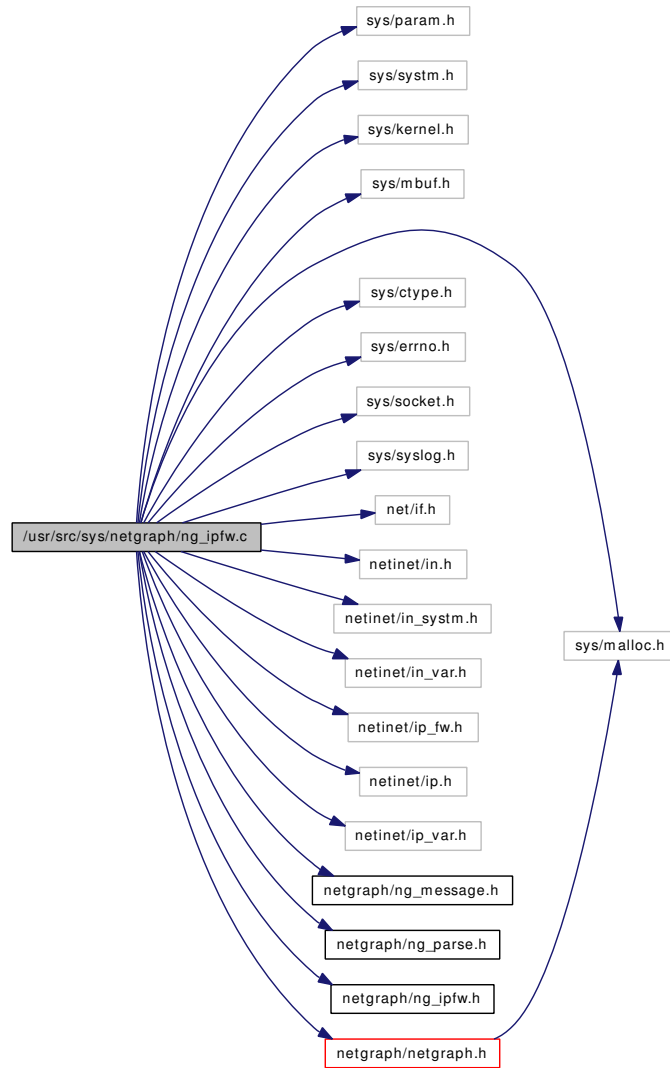
7.113.1.2 #define [NGM_IP_INPUT_COOKIE](#) 994874907

Definition at line 75 of file ng_ip_input.h.

7.114 /usr/src/sys/netgraph/ng_ipfw.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/errno.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/in_var.h>
#include <netinet/ip_fw.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_ipfw.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_ipfw.c:



Data Structures

- struct [ng_ipfw_hook_priv](#)

Typedefs

- typedef [ng_ipfw_hook_priv * hpriv_p](#)

Functions

- static int [ng_ipfw_mod_event](#) (module_t mod, int event, void *data)
- static hook_p [ng_ipfw_findhook1](#) (node_p, u_int16_t)
- static int [ng_ipfw_input](#) (struct mbuf **, int, struct ip_fw_args *, int)
- [NETGRAPH_INIT](#) (ipfw,&[ng_ipfw_tpestruct](#))
- [MODULE_DEPEND](#) (ng_ipfw, ipfw, 2, 2, 2)

- static int `ng_ipfw_constructor` (`node_p` node)
- static int `ng_ipfw_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_ipfw_connect` (`hook_p` hook)
- `hook_p` `ng_ipfw_findhook` (`node_p` node, const char *name)
- static int `ng_ipfw_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_ipfw_shutdown` (`node_p` node)
- static int `ng_ipfw_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_ipfw_constructor`
- static `ng_shutdown_t` `ng_ipfw_shutdown`
- static `ng_newhook_t` `ng_ipfw_newhook`
- static `ng_connect_t` `ng_ipfw_connect`
- static `ng_findhook_t` `ng_ipfw_findhook`
- static `ng_rcvdata_t` `ng_ipfw_rcvdata`
- static `ng_disconnect_t` `ng_ipfw_disconnect`
- static `node_p` `fw_node`
- static struct `ng_type` `ng_ipfw_typestruct`

7.114.1 Typedef Documentation

7.114.1.1 typedef struct `ng_ipfw_hook_priv*` `hpriv_p`

Definition at line 90 of file `ng_ipfw.c`.

7.114.2 Function Documentation

7.114.2.1 `MODULE_DEPEND` (`ng_ipfw`, `ipfw`, 2, 2, 2)

7.114.2.2 `NETGRAPH_INIT` (`ipfw`, & `ng_ipfw_typestruct`)

7.114.2.3 static int `ng_ipfw_connect` (`hook_p` *hook*) [static]

Definition at line 182 of file `ng_ipfw.c`.

References `ng_ipfw_hook_priv::hook`, and `NG_HOOK_FORCE_QUEUE`.

7.114.2.4 static int `ng_ipfw_constructor` (`node_p` *node*) [static]

Definition at line 137 of file `ng_ipfw.c`.

7.114.2.5 static int `ng_ipfw_disconnect` (`hook_p` *hook*) [static]

Definition at line 330 of file `ng_ipfw.c`.

References `NG_HOOK_PRIVATE`, and `NG_HOOK_SET_PRIVATE`.

7.114.2.6 `hook_p ng_ipfw_findhook (node_p node, const char * name)`

Definition at line 190 of file `ng_ipfw.c`.

References `ng_ipfw_findhook1()`.

Here is the call graph for this function:



7.114.2.7 `static hook_p ng_ipfw_findhook1 (node_p, u_int16_t) [static]`

Definition at line 203 of file `ng_ipfw.c`.

References `ng_ipfw_hook_priv::hook`, `NG_HOOK_IS_VALID`, and `NG_HOOK_PRIVATE`.

Referenced by `ng_ipfw_findhook()`, and `ng_ipfw_input()`.

7.114.2.8 `static int ng_ipfw_input (struct mbuf **, int, struct ip_fw_args *, int) [static]`

Definition at line 261 of file `ng_ipfw.c`.

References `ng_ipfw_tag::dir`, `fw_node`, `ng_ipfw_tag::ifp`, `ng_ipfw_tag::mt`, `ng_ipfw_findhook1()`, `NG_SEND_DATA_ONLY`, `NGM_IPFW_COOKIE`, `ng_ipfw_tag::rule`, and `TAGSIZ`.

Referenced by `ng_ipfw_mod_event()`.

Here is the call graph for this function:

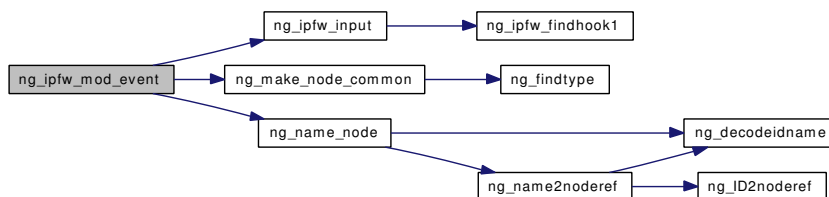


7.114.2.9 `static int ng_ipfw_mod_event (module_t mod, int event, void * data) [static]`

Definition at line 93 of file `ng_ipfw.c`.

References `fw_node`, `ng_ipfw_input()`, `ng_ipfw_input_p`, `ng_ipfw_tpestruct`, `ng_make_node_common()`, and `ng_name_node()`.

Here is the call graph for this function:



7.114.2.10 `static int ng_ipfw_newhook (node_p node, hook_p hook, const char * name)`
[static]

Definition at line 143 of file ng_ipfw.c.

References ng_ipfw_hook_priv::hook, NG_HOOK_SET_PRIVATE, and ng_ipfw_hook_priv::rulenumber.

7.114.2.11 `static int ng_ipfw_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 219 of file ng_ipfw.c.

References ng_ipfw_tag::dir, NG_FREE_ITEM, NG_FREE_M, NG_IPFW_IN, NG_IPFW_OUT, NGI_GET_M, and NGM_IPFW_COOKIE.

7.114.2.12 `static int ng_ipfw_shutdown (node_p node)` [static]

Definition at line 316 of file ng_ipfw.c.

References ng_ipfw_input_p, and NG_NODE_UNREF.

7.114.3 Variable Documentation

7.114.3.1 `node_p fw_node` [static]

Definition at line 67 of file ng_ipfw.c.

Referenced by ng_ipfw_input(), and ng_ipfw_mod_event().

7.114.3.2 `ng_connect_t ng_ipfw_connect` [static]

Definition at line 57 of file ng_ipfw.c.

7.114.3.3 `ng_constructor_t ng_ipfw_constructor` [static]

Definition at line 54 of file ng_ipfw.c.

7.114.3.4 `ng_disconnect_t ng_ipfw_disconnect` [static]

Definition at line 60 of file ng_ipfw.c.

7.114.3.5 `ng_findhook_t ng_ipfw_findhook` [static]

Definition at line 58 of file ng_ipfw.c.

7.114.3.6 `ng_newhook_t ng_ipfw_newhook` [static]

Definition at line 56 of file ng_ipfw.c.

7.114.3.7 `ng_rcvdata_t ng_ipfw_rcvdata` [static]

Definition at line 59 of file ng_ipfw.c.

7.114.3.8 `ng_shutdown_t ng_ipfw_shutdown` [static]

Definition at line 55 of file ng_ipfw.c.

7.114.3.9 `struct ng_type ng_ipfw_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_IPFW_NODE_TYPE,
    .mod_event =   ng_ipfw_mod_event,
    .constructor = ng_ipfw_constructor,
    .shutdown =    ng_ipfw_shutdown,
    .newhook =     ng_ipfw_newhook,
    .connect =     ng_ipfw_connect,
    .findhook =    ng_ipfw_findhook,
    .rcvdata =     ng_ipfw_rcvdata,
    .disconnect =  ng_ipfw_disconnect,
}
```

Definition at line 70 of file ng_ipfw.c.

Referenced by ng_ipfw_mod_event().

7.115 /usr/src/sys/netgraph/ng_ipfw.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_ipfw_tag](#)

Defines

- #define [NG_IPFW_NODE_TYPE](#) "ipfw"
- #define [NGM_IPFW_COOKIE](#) 1105988990
- #define [NG_IPFW_LOADED](#) ([ng_ipfw_input_p](#) != NULL)
- #define [NG_IPFW_OUT](#) 0
- #define [NG_IPFW_IN](#) 1
- #define [TAGSIZ](#) (sizeof(struct [ng_ipfw_tag](#)) - sizeof(struct [m_tag](#)))

Typedefs

- typedef int [ng_ipfw_input_t](#) (struct mbuf **, int, struct ip_fw_args *, int)

Variables

- [ng_ipfw_input_t](#) * [ng_ipfw_input_p](#)

7.115.1 Define Documentation

7.115.1.1 #define [NG_IPFW_IN](#) 1

Definition at line 44 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_rcvdata\(\)](#).

7.115.1.2 #define [NG_IPFW_LOADED](#) ([ng_ipfw_input_p](#) != NULL)

Definition at line 36 of file [ng_ipfw.h](#).

7.115.1.3 #define [NG_IPFW_NODE_TYPE](#) "ipfw"

Definition at line 29 of file [ng_ipfw.h](#).

7.115.1.4 #define [NG_IPFW_OUT](#) 0

Definition at line 43 of file [ng_ipfw.h](#).

Referenced by [ng_ipfw_rcvdata\(\)](#).

7.115.1.5 #define NGM_IPFW_COOKIE 1105988990

Definition at line 30 of file ng_ipfw.h.

Referenced by ng_ipfw_input(), and ng_ipfw_rcvdata().

7.115.1.6 #define TAGSIZ (sizeof(struct ng_ipfw_tag) - sizeof(struct m_tag))

Definition at line 47 of file ng_ipfw.h.

Referenced by ng_ipfw_input().

7.115.2 Typedef Documentation**7.115.2.1 typedef int ng_ipfw_input_t(struct mbuf **, int, struct ip_fw_args *, int)**

Definition at line 34 of file ng_ipfw.h.

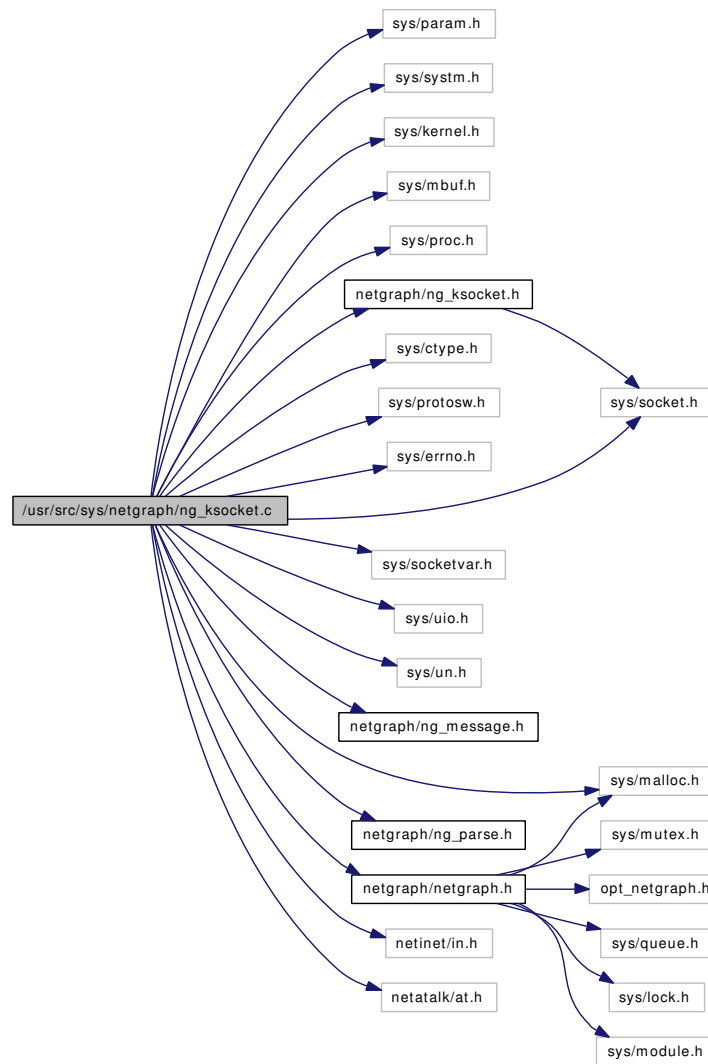
7.115.3 Variable Documentation**7.115.3.1 ng_ipfw_input_t* ng_ipfw_input_p**

Referenced by ng_ipfw_mod_event(), and ng_ipfw_shutdown().

7.116 /usr/src/sys/netgraph/ng_ksocket.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/proc.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/protosw.h>
#include <sys/errno.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/uio.h>
#include <sys/un.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_ksocket.h>
#include <netinet/in.h>
#include <netatalk/at.h>
```

Include dependency graph for ng_ksocket.c:



Data Structures

- struct [ng_ksocket_private](#)
- struct [ng_ksocket_alias](#)

Defines

- #define [M_NETGRAPH_KSOCKET](#) M_NETGRAPH
- #define [OFFSETOF](#)(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))
- #define [SADATA_OFFSET](#) (OFFSETOF(struct sockaddr, sa_data))
- #define [KSF_CONNECTING](#) 0x00000001
- #define [KSF_ACCEPTING](#) 0x00000002
- #define [KSF_EOFSEEN](#) 0x00000004
- #define [KSF_CLONED](#) 0x00000008
- #define [KSF_EMBRYONIC](#) 0x00000010
- #define [ERRROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef `ng_ksocket_private` * `priv_p`

Functions

- static int `ng_ksocket_check_accept` (`priv_p`)
- static void `ng_ksocket_finish_accept` (`priv_p`)
- static void `ng_ksocket_incoming` (struct socket *`so`, void *`arg`, int `waitflag`)
- static int `ng_ksocket_parse` (const struct `ng_ksocket_alias` *`aliases`, const char *`s`, int `family`)
- static void `ng_ksocket_incoming2` (`node_p` `node`, `hook_p` `hook`, void *`arg1`, int `waitflag`)
- static int `ng_parse_generic_sockdata_getLength` (const struct `ng_parse_type` *`type`, const u_char *`start`, const u_char *`buf`)
- static int `ng_ksocket_sockaddr_parse` (const struct `ng_parse_type` *`type`, const char *`s`, int *`off`, const u_char *`const_start`, u_char *`const_buf`, int *`buflen`)
- static int `ng_ksocket_sockaddr_unparse` (const struct `ng_parse_type` *`type`, const u_char *`data`, int *`off`, char *`cbuf`, int `cbuflen`)
- static int `ng_parse_sockoptval_getLength` (const struct `ng_parse_type` *`type`, const u_char *`start`, const u_char *`buf`)
- `NETGRAPH_INIT` (`ksocket`, &`ng_ksocket_typestruct`)
- static int `ng_ksocket_constructor` (`node_p` `node`)
- static int `ng_ksocket_newhook` (`node_p` `node`, `hook_p` `hook`, const char *`name0`)
- static int `ng_ksocket_connect` (`hook_p` `hook`)
- static int `ng_ksocket_rcvmsg` (`node_p` `node`, `item_p` `item`, `hook_p` `lasthook`)
- static int `ng_ksocket_rcvdata` (`hook_p` `hook`, `item_p` `item`)
- static int `ng_ksocket_shutdown` (`node_p` `node`)
- static int `ng_ksocket_disconnect` (`hook_p` `hook`)

Variables

- static `ng_constructor_t` `ng_ksocket_constructor`
- static `ng_rcvmsg_t` `ng_ksocket_rcvmsg`
- static `ng_shutdown_t` `ng_ksocket_shutdown`
- static `ng_newhook_t` `ng_ksocket_newhook`
- static `ng_rcvdata_t` `ng_ksocket_rcvdata`
- static `ng_connect_t` `ng_ksocket_connect`
- static `ng_disconnect_t` `ng_ksocket_disconnect`
- static struct `ng_ksocket_alias` `ng_ksocket_families` []
- static struct `ng_ksocket_alias` `ng_ksocket_types` []
- static struct `ng_ksocket_alias` `ng_ksocket_protos` []
- static struct `ng_parse_type` `ng_ksocket_generic_sockdata_type`
- static struct `ng_parse_struct_field` `ng_parse_generic_sockaddr_type_fields` []
- static struct `ng_parse_type` `ng_ksocket_generic_sockaddr_type`
- static struct `ng_parse_type` `ng_ksocket_sockaddr_type`
- static struct `ng_parse_type` `ng_ksocket_sockoptval_type`
- static struct `ng_parse_struct_field` `ng_ksocket_sockopt_type_fields` [] = `NG_KSOCKET_-SOCKOPT_INFO(&ng_ksocket_sockoptval_type)`
- static struct `ng_parse_type` `ng_ksocket_sockopt_type`
- static struct `ng_parse_struct_field` `ng_ksocket_accept_type_fields` [] = `NGM_KSOCKET_-ACCEPT_INFO`

- static struct [ng_parse_type](#) [ng_ksocket_accept_type](#)
- static struct [ng_cmdlist](#) [ng_ksocket_cmds](#) []
- static struct [ng_type](#) [ng_ksocket_tpestruct](#)

7.116.1 Define Documentation

7.116.1.1 **#define ERROUT(x) do { error = (x); goto done; } while (0)**

Definition at line 508 of file `ng_ksocket.c`.

7.116.1.2 **#define KSF_ACCEPTING 0x00000002**

Definition at line 95 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_incoming2()`, and `ng_ksocket_rcvmsg()`.

7.116.1.3 **#define KSF_CLONED 0x00000008**

Definition at line 97 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_connect()`, `ng_ksocket_finish_accept()`, and `ng_ksocket_newhook()`.

7.116.1.4 **#define KSF_CONNECTING 0x00000001**

Definition at line 94 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_incoming2()`, and `ng_ksocket_rcvmsg()`.

7.116.1.5 **#define KSF_EMBRYONIC 0x00000010**

Definition at line 98 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_finish_accept()`, `ng_ksocket_newhook()`, and `ng_ksocket_shutdown()`.

7.116.1.6 **#define KSF_EOFSEEN 0x00000004**

Definition at line 96 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_incoming2()`.

7.116.1.7 **#define M_NETGRAPH_KSOCKET M_NETGRAPH**

Definition at line 74 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_constructor()`, `ng_ksocket_shutdown()`, `ng_ksocket_sockaddr_parse()`, and `ng_ksocket_sockaddr_unparse()`.

7.116.1.8 **#define OFFSETOF(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))**

Definition at line 77 of file `ng_ksocket.c`.

7.116.1.9 #define SADATA_OFFSET (OFFSETOF(struct sockaddr, sa_data))

Definition at line 78 of file ng_ksocket.c.

Referenced by ng_ksocket_rcvmsg(), ng_ksocket_sockaddr_parse(), and ng_parse_generic_sockdata_getLength().

7.116.2 Typedef Documentation**7.116.2.1 typedef struct ng_ksocket_private* priv_p**

Definition at line 91 of file ng_ksocket.c.

7.116.3 Function Documentation**7.116.3.1 NETGRAPH_INIT (ksocket, & ng_ksocket_typestruct)****7.116.3.2 static int ng_ksocket_check_accept (priv_p) [static]**

Definition at line 1172 of file ng_ksocket.c.

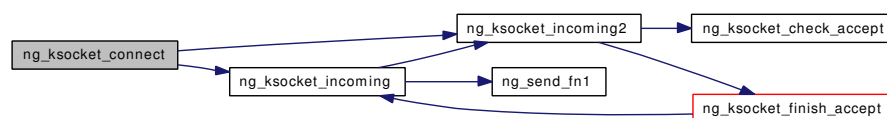
Referenced by ng_ksocket_incoming2(), and ng_ksocket_rcvmsg().

7.116.3.3 static int ng_ksocket_connect (hook_p hook) [static]

Definition at line 611 of file ng_ksocket.c.

References ng_ksocket_private::hook, KSF_CLONED, NG_HOOK_NODE, ng_ksocket_incoming(), ng_ksocket_incoming2(), NG_NODE_PRIVATE, ng_send_fn, ng_ksocket_private::node, and ng_ksocket_private::so.

Here is the call graph for this function:

**7.116.3.4 static int ng_ksocket_constructor (node_p node) [static]**

Definition at line 520 of file ng_ksocket.c.

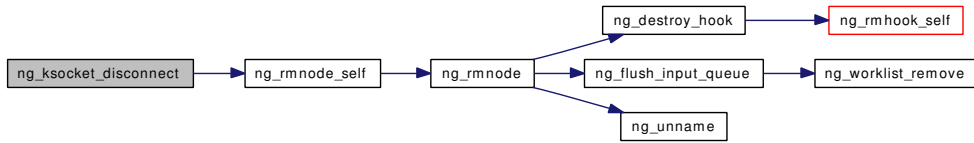
References M_NETGRAPH_KSOCKET, NG_NODE_SET_PRIVATE, and ng_ksocket_private::node.

7.116.3.5 static int ng_ksocket_disconnect (hook_p hook) [static]

Definition at line 977 of file ng_ksocket.c.

References ng_ksocket_private::hook, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, and ng_rmnode_self().

Here is the call graph for this function:



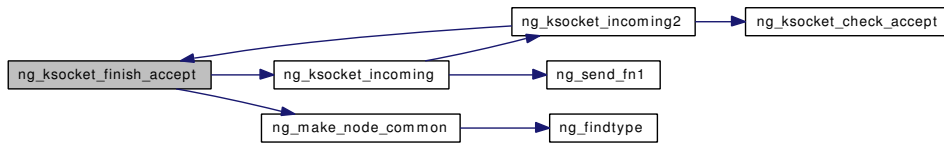
7.116.3.6 static void ng_ksocket_finish_accept (priv_p) [static]

Definition at line 1195 of file ng_ksocket.c.

References ng_ksocket_accept::addr, ng_mesg::data, ng_mesg::ng_msghdr::flags, ng_mesg::header, KSF_CLONED, KSF_EMBRYONIC, ng_ksocket_constructor, ng_ksocket_incoming(), ng_ksocket_ttypestruct, ng_make_node_common(), NG_MKMESSAGE, NG_NODE_ID, NG_NODE_PRIVATE, NG_NODE_UNREF, NG_SEND_MSG_ID, NGF_RESP, NGM_KSOCKET_ACCEPT, NGM_KSOCKET_COOKIE, ng_ksocket_private::node, ng_ksocket_accept::nodeid, OFFSETOF, ng_ksocket_private::so, and ng_mesg::ng_msghdr::token.

Referenced by ng_ksocket_incoming2(), and ng_ksocket_rcvmsg().

Here is the call graph for this function:



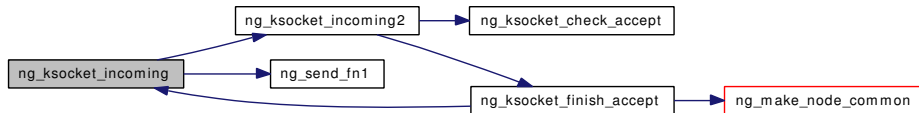
7.116.3.7 static void ng_ksocket_incoming (struct socket * so, void * arg, int waitflag) [static]

Definition at line 1012 of file ng_ksocket.c.

References ng_ksocket_incoming2(), NG_QUEUE, ng_send_fn1(), NG_WAITOK, and ng_ksocket_private::node.

Referenced by ng_ksocket_connect(), and ng_ksocket_finish_accept().

Here is the call graph for this function:



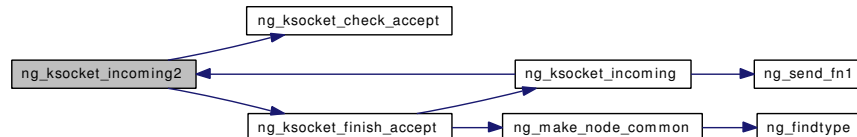
7.116.3.8 static void ng_ksocket_incoming2 (node_p node, hook_p hook, void * arg1, int waitflag) [static]

Definition at line 1035 of file ng_ksocket.c.

References `sa_tag::id`, `KSF_ACCEPTING`, `KSF_CONNECTING`, `KSF_EOFSEEN`, `ng_ID_t`, `ng_ksocket_check_accept()`, `ng_ksocket_finish_accept()`, `NG_KSOCKET_TAG_SOCKADDR`, `NG_MKMESSAGE`, `NG_NODE_ID`, `NG_NODE_NOT_VALID`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `NG_SEND_MSG_ID`, `NGF_RESP`, `NGM_KSOCKET_CONNECT`, `NGM_KSOCKET_COOKIE`, `ng_ksocket_private::node`, `sa_tag::sa`, `ng_ksocket_private::so`, and `sa_tag::tag`.

Referenced by `ng_ksocket_connect()`, and `ng_ksocket_incoming()`.

Here is the call graph for this function:



7.116.3.9 static int ng_ksocket_newhook (node_p node, hook_p hook, const char * name0) [static]

Definition at line 548 of file `ng_ksocket.c`.

References `ng_ksocket_private::hook`, `KSF_CLONED`, `KSF_EMBRYONIC`, `name`, `NG_HOOK_FORCE_QUEUE`, `NG_HOOKSIZ`, `ng_ksocket_families`, `ng_ksocket_parse()`, `ng_ksocket_protos`, `ng_ksocket_types`, `NG_NODE_PRIVATE`, and `ng_ksocket_private::node`.

Here is the call graph for this function:



7.116.3.10 static int ng_ksocket_parse (const struct ng_ksocket_alias * aliases, const char * s, int family) [static]

Definition at line 1292 of file `ng_ksocket.c`.

References `name`, and `ng_ksocket_alias::name`.

Referenced by `ng_ksocket_newhook()`, and `ng_ksocket_sockaddr_parse()`.

7.116.3.11 static int ng_ksocket_rcvdata (hook_p hook, item_p item) [static]

Definition at line 897 of file `ng_ksocket.c`.

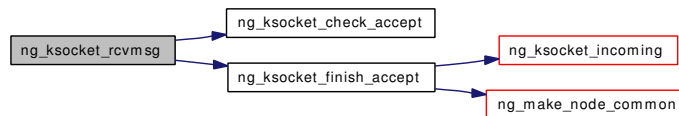
References `ng_ksocket_private::hook`, `sa_tag::id`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_KSOCKET_TAG_SOCKADDR`, `NG_NODE_ID`, `NG_NODE_PRIVATE`, `NGI_GET_M`, `NGM_KSOCKET_COOKIE`, `ng_ksocket_private::node`, `sa_tag::sa`, and `ng_ksocket_private::so`.

7.116.3.12 static int ng_ksocket_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 671 of file `ng_ksocket.c`.

References `ng_msg::ng_msghdr::arglen`, `ng_msg::ng_msghdr::cmd`, `ng_msg::data`, `ERROUT`, `ng_msg::header`, `KSF_ACCEPTING`, `KSF_CONNECTING`, `NG_FREE_MSG`, `ng_ID_t`, `ng_ksocket_check_accept()`, `ng_ksocket_finish_accept()`, `NG_KSOCKET_MAX_OPTLEN`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGI_RETADDR`, `NGM_KSOCKET_ACCEPT`, `NGM_KSOCKET_BIND`, `NGM_KSOCKET_CONNECT`, `NGM_KSOCKET_COOKIE`, `NGM_KSOCKET_GETNAME`, `NGM_KSOCKET_GETOPT`, `NGM_KSOCKET_GETPEERNAME`, `NGM_KSOCKET_LISTEN`, `NGM_KSOCKET_SETOPT`, `ng_ksocket_private::node`, `SADATA_OFFSET`, `ng_ksocket_private::so`, `ng_msg::ng_msghdr::token`, and `ng_msg::ng_msghdr::typecookie`.

Here is the call graph for this function:

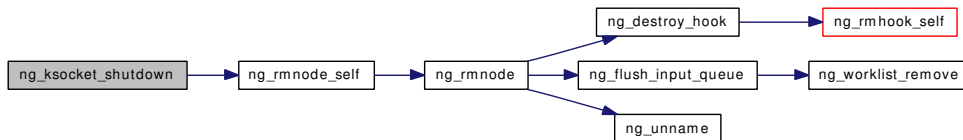


7.116.3.13 `static int ng_ksocket_shutdown (node_p node) [static]`

Definition at line 935 of file `ng_ksocket.c`.

References `KSF_EMBRYONIC`, `M_NETGRAPH_KSOCKET`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `ng_rmnode_self()`, and `ng_ksocket_private::node`.

Here is the call graph for this function:

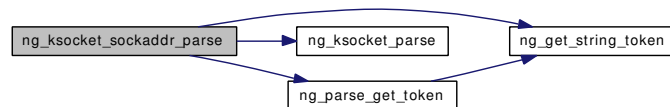


7.116.3.14 `static int ng_ksocket_sockaddr_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen) [static]`

Definition at line 204 of file `ng_ksocket.c`.

References `M_NETGRAPH_KSOCKET`, `ng_get_string_token()`, `ng_ksocket_families`, `ng_ksocket_generic_sockaddr_type`, `ng_ksocket_parse()`, `ng_parse_get_token()`, `OFFSETOF`, `ng_parse_type::parse`, `SADATA_OFFSET`, `ng_parse_type::supertype`, and `T_LBRACE`.

Here is the call graph for this function:

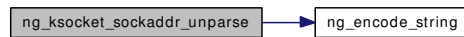


7.116.3.15 `static int ng_ksocket_sockaddr_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 317 of file ng_ksocket.c.

References `M_NETGRAPH_KSOCKET`, `ng_encode_string()`, `ng_ksocket_generic_sockaddr_type`, `OFFSETOF`, `ng_parse_type::supertype`, and `ng_parse_type::unparse`.

Here is the call graph for this function:



7.116.3.16 `static int ng_parse_generic_sockdata_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 172 of file ng_ksocket.c.

References `SADATA_OFFSET`.

7.116.3.17 `static int ng_parse_sockoptval_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 395 of file ng_ksocket.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::header`, and `OFFSETOF`.

7.116.4 Variable Documentation

7.116.4.1 `struct ng_parse_type ng_ksocket_accept_type` [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_ksocket_accept_type_fields
}
  
```

Definition at line 427 of file ng_ksocket.c.

7.116.4.2 `struct ng_parse_struct_field ng_ksocket_accept_type_fields[] = NGM_KSOCKET_ACCEPT_INFO` [static]

Definition at line 426 of file ng_ksocket.c.

7.116.4.3 `struct ng_cmdlist ng_ksocket_cmds[]` [static]

Definition at line 433 of file ng_ksocket.c.

7.116.4.4 `ng_connect_t ng_ksocket_connect` [static]

Definition at line 106 of file `ng_ksocket.c`.

7.116.4.5 `ng_constructor_t ng_ksocket_constructor` [static]

Definition at line 101 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_finish_accept()`.

7.116.4.6 `ng_disconnect_t ng_ksocket_disconnect` [static]

Definition at line 107 of file `ng_ksocket.c`.

7.116.4.7 `struct ng_ksocket_alias ng_ksocket_families[]` [static]**Initial value:**

```
{
    { "local",      PF_LOCAL      },
    { "inet",      PF_INET       },
    { "inet6",     PF_INET6      },
    { "atalk",     PF_APPLETALK  },
    { "ipx",      PF_IPX        },
    { "atm",      PF_ATM        },
    { NULL,       -1            },
}
```

Definition at line 117 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_newhook()`, and `ng_ksocket_sockaddr_parse()`.

7.116.4.8 `struct ng_parse_type ng_ksocket_generic_sockaddr_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_parse_generic_sockaddr_type_fields
}
```

Definition at line 195 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_sockaddr_parse()`, and `ng_ksocket_sockaddr_unparse()`.

7.116.4.9 `struct ng_parse_type ng_ksocket_generic_sockdata_type` [static]**Initial value:**

```
{
    &ng_parse_bytearray_type,
    &ng_parse_generic_sockdata_getLength
}
```

Definition at line 182 of file `ng_ksocket.c`.

7.116.4.10 `ng_newhook_t ng_ksocket_newhook` [static]

Definition at line 104 of file `ng_ksocket.c`.

7.116.4.11 `struct ng_ksocket_alias ng_ksocket_protos[]` [static]

Initial value:

```
{
    { "ip",          IPPROTO_IP,          PF_INET          },
    { "raw",        IPPROTO_RAW,        PF_INET          },
    { "icmp",       IPPROTO_ICMP,       PF_INET          },
    { "igmp",       IPPROTO_IGMP,       PF_INET          },
    { "tcp",        IPPROTO_TCP,        PF_INET          },
    { "udp",        IPPROTO_UDP,        PF_INET          },
    { "gre",        IPPROTO_GRE,        PF_INET          },
    { "esp",        IPPROTO_ESP,        PF_INET          },
    { "ah",         IPPROTO_AH,         PF_INET          },
    { "swipe",      IPPROTO_SWIPE,      PF_INET          },
    { "encap",      IPPROTO_ENCAP,      PF_INET          },
    { "divert",     IPPROTO_DIVERT,     PF_INET          },
    { "pim",        IPPROTO_PIM,        PF_INET          },
    { "ddp",        IPPROTO_DDP,        PF_APPLETALK    },
    { "arp",        IPPROTO_ARP,        PF_APPLETALK    },
    { NULL,         -1,

```

Definition at line 138 of file `ng_ksocket.c`.

Referenced by `ng_ksocket_newhook()`.

7.116.4.12 `ng_rcvdata_t ng_ksocket_rcvdata` [static]

Definition at line 105 of file `ng_ksocket.c`.

7.116.4.13 `ng_rcvmsg_t ng_ksocket_rcvmsg` [static]

Definition at line 102 of file `ng_ksocket.c`.

7.116.4.14 `ng_shutdown_t ng_ksocket_shutdown` [static]

Definition at line 103 of file `ng_ksocket.c`.

7.116.4.15 `struct ng_parse_type ng_ksocket_sockaddr_type` [static]

Initial value:

```
{
    NULL,
    NULL,
    NULL,
    &ng_ksocket_sockaddr_parse,
    &ng_ksocket_sockaddr_unparse,
    NULL
}
```

Definition at line 378 of file `ng_ksocket.c`.

7.116.4.16 `struct ng_parse_type ng_ksocket_sockopt_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_ksocket_sockopt_type_fields
}
```

Definition at line 419 of file ng_ksocket.c.

7.116.4.17 `struct ng_parse_struct_field ng_ksocket_sockopt_type_fields[] = NG_KSOCKET_SOCKOPT_INFO(&ng_ksocket_sockoptval_type)` [static]

Definition at line 418 of file ng_ksocket.c.

7.116.4.18 `struct ng_parse_type ng_ksocket_sockoptval_type` [static]**Initial value:**

```
{
    &ng_parse_bytearray_type,
    &ng_parse_sockoptval_getLength
}
```

Definition at line 411 of file ng_ksocket.c.

7.116.4.19 `struct ng_ksocket_alias ng_ksocket_types[]` [static]**Initial value:**

```
{
    { "stream",      SOCK_STREAM   },
    { "dgram",      SOCK_DGRAM    },
    { "raw",        SOCK_RAW      },
    { "rdm",        SOCK_RDM      },
    { "seqpacket",  SOCK_SEQPACKET },
    { NULL,        -1             },
}
```

Definition at line 128 of file ng_ksocket.c.

Referenced by ng_ksocket_newhook().

7.116.4.20 `struct ng_type ng_ksocket_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_KSOCKET_NODE_TYPE,
    .constructor = ng_ksocket_constructor,
    .rcvmsg =      ng_ksocket_rcvmsg,
    .shutdown =    ng_ksocket_shutdown,
}
```

```
.newhook =      ng_ksocket_newhook,  
.connect =     ng_ksocket_connect,  
.rcvdata =     ng_ksocket_rcvdata,  
.disconnect =  ng_ksocket_disconnect,  
.cmdlist =     ng_ksocket_cmds,  
}
```

Definition at line 494 of file ng_ksocket.c.

Referenced by ng_ksocket_finish_accept().

7.116.4.21 struct ng_parse_struct_field ng_parse_generic_sockaddr_type_fields[] [static]

Initial value:

```
{  
    { "len",      &ng_parse_uint8_type      },  
    { "family",  &ng_parse_uint8_type      },  
    { "data",    &ng_ksocket_generic_sockdata_type },  
    { NULL }  
}
```

Definition at line 189 of file ng_ksocket.c.

7.117 /usr/src/sys/netgraph/ng_ksocket.h File Reference

```
#include <sys/socket.h>
```

Include dependency graph for ng_ksocket.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_ksocket_sockopt](#)
- struct [ng_ksocket_accept](#)
- struct [sa_tag](#)

Defines

- #define [NG_KSOCKET_NODE_TYPE](#) "ksocket"
- #define [NGM_KSOCKET_COOKIE](#) 942710669
- #define [NG_KSOCKET_MAX_OPTLEN](#) 1024
- #define [NG_KSOCKET_SOCKOPT_INFO](#)(svtype)
- #define [NGM_KSOCKET_ACCEPT_INFO](#)
- #define [NG_KSOCKET_TAG_SOCKADDR](#) 1

Enumerations

- enum {
 - [NGM_KSOCKET_BIND](#) = 1, [NGM_KSOCKET_LISTEN](#), [NGM_KSOCKET_ACCEPT](#), [NGM_KSOCKET_CONNECT](#),
 - [NGM_KSOCKET_GETNAME](#), [NGM_KSOCKET_GETPEERNAME](#), [NGM_KSOCKET_SETOPT](#), [NGM_KSOCKET_GETOPT](#) }

7.117.1 Define Documentation

7.117.1.1 #define NG_KSOCKET_MAX_OPTLEN 1024

Definition at line 63 of file ng_ksocket.h.

Referenced by [ng_ksocket_rcvmsg\(\)](#).

7.117.1.2 #define NG_KSOCKET_NODE_TYPE "ksocket"

Definition at line 50 of file ng_ksocket.h.

7.117.1.3 #define NG_KSOCKET_SOCKETOPT_INFO(svtype)**Value:**

```
{
    { "level",          \      &ng_parse_int32_type    }, \
    { "name",          \      &ng_parse_int32_type    }, \
    { "value",        \      (svtype)                }, \
    { NULL }
}
```

Definition at line 66 of file ng_ksocket.h.

7.117.1.4 #define NG_KSOCKET_TAG_SOCKADDR 1

Definition at line 108 of file ng_ksocket.h.

Referenced by ng_ksocket_incoming2(), and ng_ksocket_rcvdata().

7.117.1.5 #define NGM_KSOCKET_ACCEPT_INFO**Value:**

```
{
    { "nodeid",        \      &ng_parse_hint32_type    }, \
    { "addr",         \      &ng_ksocket_generic_sockaddr_type }, \
    { NULL }
}
```

Definition at line 80 of file ng_ksocket.h.

7.117.1.6 #define NGM_KSOCKET_COOKIE 942710669

Definition at line 51 of file ng_ksocket.h.

Referenced by ng_ksocket_finish_accept(), ng_ksocket_incoming2(), ng_ksocket_rcvdata(), and ng_ksocket_rcvmsg().

7.117.2 Enumeration Type Documentation**7.117.2.1 anonymous enum****Enumerator:**

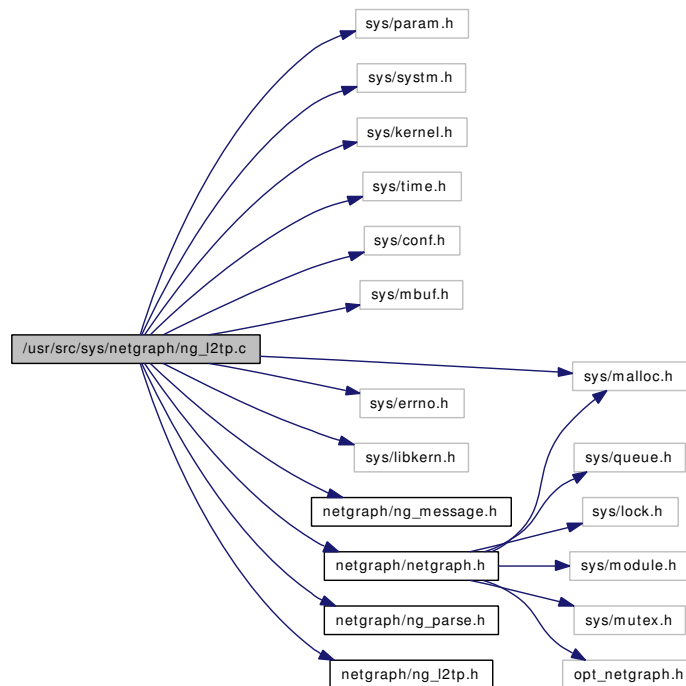
NGM_KSOCKET_BIND
NGM_KSOCKET_LISTEN
NGM_KSOCKET_ACCEPT
NGM_KSOCKET_CONNECT
NGM_KSOCKET_GETNAME
NGM_KSOCKET_GETPEERNAME
NGM_KSOCKET_SETOPT
NGM_KSOCKET_GETOPT

Definition at line 87 of file ng_ksocket.h.

7.118 /usr/src/sys/netgraph/ng_l2tp.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/time.h>
#include <sys/conf.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/libkern.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_l2tp.h>
```

Include dependency graph for ng_l2tp.c:



Data Structures

- struct [l2tp_seq](#)
- struct [ng_l2tp_private](#)
- struct [ng_l2tp_hook_private](#)

Defines

- #define `M_NETGRAPH_L2TP` `M_NETGRAPH`
- #define `L2TP_HDR_CTRL` `0x8000`
- #define `L2TP_HDR_LEN` `0x4000`
- #define `L2TP_HDR_SEQ` `0x0800`
- #define `L2TP_HDR_OFF` `0x0200`
- #define `L2TP_HDR_PRIO` `0x0100`
- #define `L2TP_HDR_VERS_MASK` `0x000f`
- #define `L2TP_HDR_VERSION` `0x0002`
- #define `L2TP_CTRL_0BITS` `0x030d`
- #define `L2TP_CTRL_1BITS` `0xc802`
- #define `L2TP_DATA_0BITS` `0x800d`
- #define `L2TP_DATA_1BITS` `0x0002`
- #define `L2TP_CTRL_HDR`
- #define `L2TP_DATA_HDR` (`L2TP_HDR_VERSION`)
- #define `L2TP_MAX_XWIN` `16`
- #define `L2TP_MAX_REXMIT` `5`
- #define `L2TP_MAX_REXMIT_TO` `30`
- #define `L2TP_DELAYED_ACK` `((hz + 19) / 20)`
- #define `L2TP_CONTROL_DSEQ` `1`
- #define `L2TP_ENABLE_DSEQ` `1`
- #define `L2TP_SEQ_DIFF(x, y)` `((int)((int16_t)(x) - (int16_t)(y)))`
- #define `L2TP_SEQ_CHECK(x)` `do { } while (0)`
- #define `memmove(d, s, l)` `bcopy(s, d, l)`
- #define `L2TP_COPY_MBUF` `m_copypacket`

Typedefs

- typedef `ng_l2tp_private` * `priv_p`
- typedef `ng_l2tp_hook_private` * `hookpriv_p`

Functions

- static int `ng_l2tp_recv_lower` (`node_p` node, `item_p` item)
- static int `ng_l2tp_recv_ctrl` (`node_p` node, `item_p` item)
- static int `ng_l2tp_recv_data` (`node_p` node, `item_p` item, `hookpriv_p` hpriv)
- static int `ng_l2tp_xmit_ctrl` (`priv_p` priv, struct mbuf *m, u_int16_t ns)
- static void `ng_l2tp_seq_init` (`priv_p` priv)
- static int `ng_l2tp_seq_set` (`priv_p` priv, const struct `ng_l2tp_seq_config` *conf)
- static int `ng_l2tp_seq_adjust` (`priv_p` priv, const struct `ng_l2tp_config` *conf)
- static void `ng_l2tp_seq_reset` (`priv_p` priv)
- static void `ng_l2tp_seq_failure` (`priv_p` priv)
- static void `ng_l2tp_seq_recv_nr` (`priv_p` priv, u_int16_t nr)
- static int `ng_l2tp_seq_recv_ns` (`priv_p` priv, u_int16_t ns)
- static void `ng_l2tp_seq_xack_timeout` (`node_p` node, `hook_p` hook, void *arg1, int arg2)
- static void `ng_l2tp_seq_rack_timeout` (`node_p` node, `hook_p` hook, void *arg1, int arg2)
- `NETGRAPH_INIT` (`l2tp`, &`ng_l2tp_tpestruct`)
- static int `ng_l2tp_constructor` (`node_p` node)

- static int `ng_l2tp_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_l2tp_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_l2tp_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_l2tp_shutdown` (`node_p` node)
- static int `ng_l2tp_disconnect` (`hook_p` hook)
- static int `ng_l2tp_find_session` (`hook_p` hook, void *arg)
- static int `ng_l2tp_reset_session` (`hook_p` hook, void *arg)

Variables

- static `ng_constructor_t` `ng_l2tp_constructor`
- static `ng_rcvmsg_t` `ng_l2tp_rcvmsg`
- static `ng_shutdown_t` `ng_l2tp_shutdown`
- static `ng_newhook_t` `ng_l2tp_newhook`
- static `ng_rcvdata_t` `ng_l2tp_rcvdata`
- static `ng_disconnect_t` `ng_l2tp_disconnect`
- static `ng_fn_eachhook` `ng_l2tp_find_session`
- static `ng_fn_eachhook` `ng_l2tp_reset_session`
- static struct `ng_parse_struct_field` `ng_l2tp_seq_config_fields` [] = NG_L2TP_SEQ_CONFIG_TYPE_INFO
- static struct `ng_parse_type` `ng_l2tp_seq_config_type`
- static struct `ng_parse_struct_field` `ng_l2tp_config_type_fields` [] = NG_L2TP_CONFIG_TYPE_INFO
- static struct `ng_parse_type` `ng_l2tp_config_type`
- static struct `ng_parse_struct_field` `ng_l2tp_sess_config_type_fields` [] = NG_L2TP_SESS_CONFIG_TYPE_INFO
- static struct `ng_parse_type` `ng_l2tp_sess_config_type`
- static struct `ng_parse_struct_field` `ng_l2tp_stats_type_fields` [] = NG_L2TP_STATS_TYPE_INFO
- static struct `ng_parse_type` `ng_l2tp_stats_type`
- static struct `ng_parse_struct_field` `ng_l2tp_session_stats_type_fields` [] = NG_L2TP_SESSION_STATS_TYPE_INFO
- static struct `ng_parse_type` `ng_l2tp_session_stats_type`
- static struct `ng_cmdlist` `ng_l2tp_cmdlist` []
- static struct `ng_type` `ng_l2tp_tpestruct`

7.118.1 Define Documentation

7.118.1.1 #define L2TP_CONTROL_DSEQ 1

Definition at line 97 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_newhook()`.

7.118.1.2 #define L2TP_COPY_MBUF m_copypacket

Definition at line 346 of file `ng_l2tp.c`.

Referenced by `ng_l2tp_rcv_ctrl()`, and `ng_l2tp_seq_rack_timeout()`.

7.118.1.3 #define L2TP_CTRL_0BITS 0x030d

Definition at line 80 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.4 #define L2TP_CTRL_1BITS 0xc802

Definition at line 81 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.5 #define L2TP_CTRL_HDR**Value:**

```
(L2TP_HDR_CTRL | L2TP_HDR_LEN \
                                     | L2TP_HDR_SEQ | L2TP_HDR_VERSION)
```

Definition at line 86 of file ng_l2tp.c.

Referenced by ng_l2tp_xmit_ctrl().

7.118.1.6 #define L2TP_DATA_0BITS 0x800d

Definition at line 82 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.7 #define L2TP_DATA_1BITS 0x0002

Definition at line 83 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.8 #define L2TP_DATA_HDR (L2TP_HDR_VERSION)

Definition at line 88 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_data().

7.118.1.9 #define L2TP_DELAYED_ACK ((hz + 19) / 20)

Definition at line 94 of file ng_l2tp.c.

Referenced by ng_l2tp_seq_rcv_ns().

7.118.1.10 #define L2TP_ENABLE_DSEQ 1

Definition at line 98 of file ng_l2tp.c.

Referenced by ng_l2tp_newhook().

7.118.1.11 #define L2TP_HDR_CTRL 0x8000

Definition at line 71 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.12 #define L2TP_HDR_LEN 0x4000

Definition at line 72 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_data(), and ng_l2tp_rcv_lower().

7.118.1.13 #define L2TP_HDR_OFF 0x0200

Definition at line 74 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower().

7.118.1.14 #define L2TP_HDR_PRIO 0x0100

Definition at line 75 of file ng_l2tp.c.

7.118.1.15 #define L2TP_HDR_SEQ 0x0800

Definition at line 73 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_data(), and ng_l2tp_rcv_lower().

7.118.1.16 #define L2TP_HDR_VERS_MASK 0x000f

Definition at line 76 of file ng_l2tp.c.

7.118.1.17 #define L2TP_HDR_VERSION 0x0002

Definition at line 77 of file ng_l2tp.c.

7.118.1.18 #define L2TP_MAX_REXMIT 5

Definition at line 92 of file ng_l2tp.c.

Referenced by ng_l2tp_constructor().

7.118.1.19 #define L2TP_MAX_REXMIT_TO 30

Definition at line 93 of file ng_l2tp.c.

Referenced by ng_l2tp_constructor().

7.118.1.20 #define L2TP_MAX_XWIN 16

Definition at line 91 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_ctrl(), ng_l2tp_seq_adjust(), ng_l2tp_seq_init(), ng_l2tp_seq_rcv_nr(), and ng_l2tp_seq_reset().

7.118.1.21 #define L2TP_SEQ_CHECK(x) do { } while (0)

Definition at line 339 of file ng_l2tp.c.

Referenced by ng_l2tp_rcvdata(), ng_l2tp_seq_init(), ng_l2tp_seq_rack_timeout(), ng_l2tp_seq_reset(), ng_l2tp_seq_xack_timeout(), and ng_l2tp_shutdown().

7.118.1.22 #define L2TP_SEQ_DIFF(x, y) ((int)((int16_t)(x) - (int16_t)(y)))

Definition at line 101 of file ng_l2tp.c.

Referenced by ng_l2tp_rcv_lower(), ng_l2tp_seq_rcv_nr(), and ng_l2tp_seq_rcv_ns().

7.118.1.23 #define M_NETGRAPH_L2TP M_NETGRAPH

Definition at line 67 of file ng_l2tp.c.

Referenced by ng_l2tp_constructor(), ng_l2tp_disconnect(), ng_l2tp_newhook(), and ng_l2tp_shutdown().

7.118.1.24 #define memmove(d, s, l) bcopy(s, d, l)

Definition at line 343 of file ng_l2tp.c.

7.118.2 Typedef Documentation**7.118.2.1 typedef struct [ng_l2tp_hook_private](#)* [hookpriv_p](#)**

Definition at line 154 of file ng_l2tp.c.

7.118.2.2 typedef struct [ng_l2tp_private](#)* [priv_p](#)

Definition at line 145 of file ng_l2tp.c.

7.118.3 Function Documentation**7.118.3.1 NETGRAPH_INIT (l2tp, & [ng_l2tp_tpestruct](#))****7.118.3.2 static int [ng_l2tp_constructor](#) ([node_p node](#)) [static]**

Definition at line 356 of file ng_l2tp.c.

References L2TP_MAX_REXMIT, L2TP_MAX_REXMIT_TO, M_NETGRAPH_L2TP, ng_l2tp_seq_init(), and NG_NODE_SET_PRIVATE.

Here is the call graph for this function:

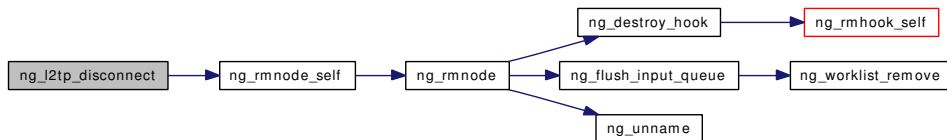


7.118.3.3 static int ng_l2tp_disconnect ([hook_p hook](#)) [static]

Definition at line 729 of file ng_l2tp.c.

References M_NETGRAPH_L2TP, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.118.3.4 static int ng_l2tp_find_session ([hook_p hook](#), void *arg) [static]

Definition at line 758 of file ng_l2tp.c.

References ng_l2tp_hook_private::conf, NG_HOOK_PRIVATE, and ng_l2tp_sess_config::session_id.

7.118.3.5 static int ng_l2tp_newhook ([node_p node](#), [hook_p hook](#), const char *name) [static]

Definition at line 383 of file ng_l2tp.c.

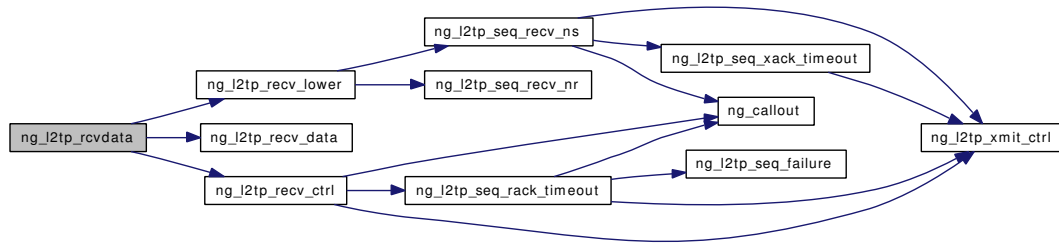
References L2TP_CONTROL_DSEQ, L2TP_ENABLE_DSEQ, M_NETGRAPH_L2TP, NG_HOOK_SET_PRIVATE, NG_L2TP_HOOK_CTRL, NG_L2TP_HOOK_LOWER, NG_L2TP_HOOK_SESSION_P, and NG_NODE_PRIVATE.

7.118.3.6 static int ng_l2tp_rcvdata ([hook_p hook](#), [item_p item](#)) [static]

Definition at line 663 of file ng_l2tp.c.

References L2TP_SEQ_CHECK, NG_FREE_ITEM, NG_HOOK_NODE, NG_HOOK_PRIVATE, ng_l2tp_rcv_ctrl(), ng_l2tp_rcv_data(), ng_l2tp_rcv_lower(), and NG_NODE_PRIVATE.

Here is the call graph for this function:

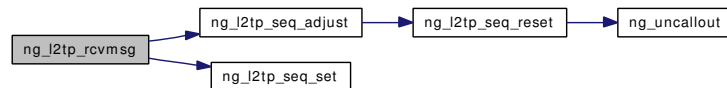


7.118.3.7 static int ng_l2tp_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 437 of file ng_l2tp.c.

References ng_msg::ng_msghdr::arglen, ng_msg::ng_msghdr::cmd, ng_l2tp_hook_private::conf, ng_msg::data, ng_msg::header, NG_FREE_MSG, NG_HOOK_PRIVATE, ng_l2tp_find_session, ng_l2tp_seq_adjust(), ng_l2tp_seq_set(), NG_MKRESPONSE, NG_NODE_FOREACH_HOOK, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGI_RETADDR, NGM_L2TP_CLR_SESSION_STATS, NGM_L2TP_CLR_STATS, NGM_L2TP_COOKIE, NGM_L2TP_GET_CONFIG, NGM_L2TP_GET_SESS_CONFIG, NGM_L2TP_GET_SESSION_STATS, NGM_L2TP_GET_STATS, NGM_L2TP_GETCLR_SESSION_STATS, NGM_L2TP_GETCLR_STATS, NGM_L2TP_SET_CONFIG, NGM_L2TP_SET_SEQ, NGM_L2TP_SET_SESS_CONFIG, ng_l2tp_sess_config::peer_id, ng_l2tp_sess_config::session_id, ng_l2tp_hook_private::stats, and ng_msg::ng_msghdr::typecookie.

Here is the call graph for this function:



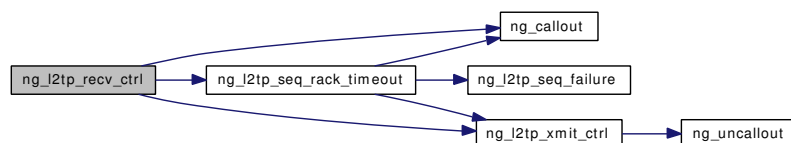
7.118.3.8 static int ng_l2tp_rcv_ctrl (node_p node, item_p item) [static]

Definition at line 1015 of file ng_l2tp.c.

References l2tp_seq::cwnd, L2TP_COPY_MBUF, L2TP_MAX_XWIN, ng_callout(), NG_FREE_ITEM, ng_l2tp_seq_rack_timeout(), ng_l2tp_xmit_ctrl(), NG_NODE_PRIVATE, NGI_GET_M, l2tp_seq::ns, l2tp_seq::rack_timer, and l2tp_seq::xwin.

Referenced by ng_l2tp_rcvdata().

Here is the call graph for this function:



7.118.3.9 static int ng_l2tp_rcv_data (node_p node, item_p item, hookpriv_p hpriv) [static]

Definition at line 1077 of file ng_l2tp.c.

References ng_l2tp_hook_private::conf, ng_l2tp_sess_config::enable_dseq, ng_l2tp_sess_config::include_length, L2TP_DATA_HDR, L2TP_HDR_LEN, L2TP_HDR_SEQ, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_NODE_PRIVATE, NGI_GET_M, ng_l2tp_hook_private::nr, ng_l2tp_hook_private::ns, ng_l2tp_sess_config::peer_id, ng_l2tp_hook_private::stats, ng_l2tp_session_stats::xmitOctets, and ng_l2tp_session_stats::xmitPackets.

Referenced by ng_l2tp_rcvdata().

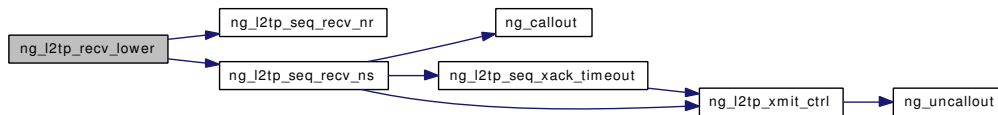
7.118.3.10 static int ng_l2tp_rcv_lower (node_p node, item_p item) [static]

Definition at line 790 of file ng_l2tp.c.

References ng_l2tp_hook_private::conf, ng_l2tp_sess_config::control_dseq, ng_l2tp_sess_config::enable_dseq, L2TP_CTRL_0BITS, L2TP_CTRL_1BITS, L2TP_DATA_0BITS, L2TP_DATA_1BITS, L2TP_HDR_CTRL, L2TP_HDR_LEN, L2TP_HDR_OFF, L2TP_HDR_SEQ, L2TP_SEQ_DIFF, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_PRIVATE, ng_l2tp_find_session, ng_l2tp_seq_rcv_nr(), ng_l2tp_seq_rcv_ns(), NG_NODE_FOREACH_HOOK, NG_NODE_PRIVATE, NGI_GET_M, ng_l2tp_hook_private::nr, ng_l2tp_session_stats::rcvOctets, ng_l2tp_session_stats::rcvPackets, and ng_l2tp_hook_private::stats.

Referenced by ng_l2tp_rcvdata().

Here is the call graph for this function:

**7.118.3.11 static int ng_l2tp_reset_session (hook_p hook, void * arg) [static]**

Definition at line 772 of file ng_l2tp.c.

References ng_l2tp_hook_private::conf, ng_l2tp_sess_config::control_dseq, ng_l2tp_sess_config::enable_dseq, NG_HOOK_PRIVATE, ng_l2tp_hook_private::nr, and ng_l2tp_hook_private::ns.

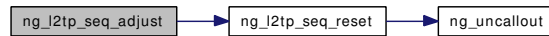
7.118.3.12 static int ng_l2tp_seq_adjust (priv_p priv, const struct ng_l2tp_config * conf) [static]

Definition at line 1201 of file ng_l2tp.c.

References ng_l2tp_config::enabled, L2TP_MAX_XWIN, l2tp_seq::max_rexmit_to, l2tp_seq::max_rexmits, ng_l2tp_seq_reset(), ng_l2tp_config::peer_win, ng_l2tp_config::rexmit_max, ng_l2tp_config::rexmit_max_to, and l2tp_seq::wmax.

Referenced by ng_l2tp_rcvmsg().

Here is the call graph for this function:



7.118.3.13 static void ng_l2tp_seq_failure (priv_p priv) [static]

Definition at line 1134 of file ng_l2tp.c.

References NG_MKMESSAGE, NG_SEND_MSG_ID, NGM_L2TP_ACK_FAILURE, and NGM_L2TP_COOKIE.

Referenced by ng_l2tp_seq_rack_timeout().

7.118.3.14 static void ng_l2tp_seq_init (priv_p priv) [static]

Definition at line 1153 of file ng_l2tp.c.

References L2TP_MAX_XWIN, L2TP_SEQ_CHECK, and ng_callout_init.

Referenced by ng_l2tp_constructor().

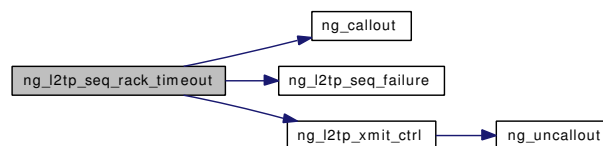
7.118.3.15 static void ng_l2tp_seq_rack_timeout (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 1418 of file ng_l2tp.c.

References l2tp_seq::acks, l2tp_seq::cwnd, L2TP_COPY_MBUF, L2TP_SEQ_CHECK, l2tp_seq::max_rexmit_to, l2tp_seq::max_rexmits, ng_callout(), ng_l2tp_seq_failure(), ng_l2tp_xmit_ctrl(), NG_NODE_PRIVATE, l2tp_seq::ns, l2tp_seq::rack, l2tp_seq::rack_timer, l2tp_seq::rexmits, l2tp_seq::ssth, and l2tp_seq::xwin.

Referenced by ng_l2tp_rcv_ctrl().

Here is the call graph for this function:



7.118.3.16 static void ng_l2tp_seq_rcv_nr (priv_p priv, u_int16_t nr) [static]

Definition at line 1271 of file ng_l2tp.c.

References l2tp_seq::cwnd, L2TP_MAX_XWIN, L2TP_SEQ_DIFF, memmove, l2tp_seq::ns, l2tp_seq::rack, l2tp_seq::rexmits, l2tp_seq::ssth, l2tp_seq::wmax, and l2tp_seq::xwin.

Referenced by ng_l2tp_rcv_lower().

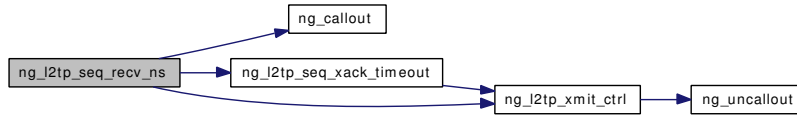
7.118.3.17 static int ng_l2tp_seq_rcv_ns (priv_p priv, u_int16_t ns) [static]

Definition at line 1366 of file ng_l2tp.c.

References L2TP_DELAYED_ACK, L2TP_SEQ_DIFF, ng_callout(), ng_l2tp_seq_xack_timeout(), ng_l2tp_xmit_ctrl(), l2tp_seq::nr, l2tp_seq::ns, and l2tp_seq::xack_timer.

Referenced by ng_l2tp_rcv_lower().

Here is the call graph for this function:



7.118.3.18 static void ng_l2tp_seq_reset (priv_p priv) [static]

Definition at line 1234 of file ng_l2tp.c.

References L2TP_MAX_XWIN, L2TP_SEQ_CHECK, ng_l2tp_reset_session, NG_NODE_FOREACH_HOOK, ng_unccallout(), l2tp_seq::rack_timer, l2tp_seq::xack_timer, and l2tp_seq::xwin.

Referenced by ng_l2tp_seq_adjust(), and ng_l2tp_shutdown().

Here is the call graph for this function:



7.118.3.19 static int ng_l2tp_seq_set (priv_p priv, const struct ng_l2tp_seq_config * conf) [static]

Definition at line 1176 of file ng_l2tp.c.

References l2tp_seq::nr, ng_l2tp_seq_config::nr, l2tp_seq::ns, ng_l2tp_seq_config::ns, l2tp_seq::rack, ng_l2tp_seq_config::rack, l2tp_seq::xack, and ng_l2tp_seq_config::xack.

Referenced by ng_l2tp_rcvmsg().

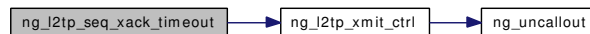
7.118.3.20 static void ng_l2tp_seq_xack_timeout (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 1397 of file ng_l2tp.c.

References L2TP_SEQ_CHECK, ng_l2tp_xmit_ctrl(), NG_NODE_PRIVATE, l2tp_seq::nr, l2tp_seq::ns, and l2tp_seq::xack.

Referenced by ng_l2tp_seq_rcv_ns().

Here is the call graph for this function:

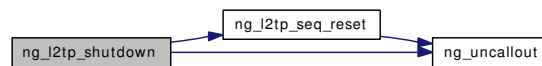


7.118.3.21 `static int ng_l2tp_shutdown (node_p node)` [static]

Definition at line 703 of file ng_l2tp.c.

References L2TP_SEQ_CHECK, M_NETGRAPH_L2TP, ng_l2tp_seq_reset(), NG_NODE_PRIVATE, NG_NODE_UNREF, ng_uncallout(), l2tp_seq::rack_timer, and l2tp_seq::xack_timer.

Here is the call graph for this function:

**7.118.3.22** `static int ng_l2tp_xmit_ctrl (priv_p priv, struct mbuf * m, u_int16_t ns)` [static]

Definition at line 1466 of file ng_l2tp.c.

References L2TP_CTRL_HDR, NG_SEND_DATA_ONLY, ng_uncallout(), l2tp_seq::nr, l2tp_seq::xack, and l2tp_seq::xack_timer.

Referenced by ng_l2tp_rcv_ctrl(), ng_l2tp_seq_rack_timeout(), ng_l2tp_seq_rcv_ns(), and ng_l2tp_seq_xack_timeout().

Here is the call graph for this function:

**7.118.4 Variable Documentation****7.118.4.1** `struct ng_cmdlist ng_l2tp_cmdlist[]` [static]

Definition at line 233 of file ng_l2tp.c.

7.118.4.2 `struct ng_parse_type ng_l2tp_config_type` [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_l2tp_config_type_fields,
}
  
```

Definition at line 203 of file ng_l2tp.c.

7.118.4.3 `struct ng_parse_struct_field ng_l2tp_config_type_fields[] = NG_L2TP_CONFIG_TYPE_INFO` [static]

Definition at line 202 of file ng_l2tp.c.

7.118.4.4 [ng_constructor_t](#) [ng_l2tp_constructor](#) [static]

Definition at line 157 of file ng_l2tp.c.

7.118.4.5 [ng_disconnect_t](#) [ng_l2tp_disconnect](#) [static]

Definition at line 162 of file ng_l2tp.c.

7.118.4.6 [ng_fn_eachhook](#) [ng_l2tp_find_session](#) [static]

Definition at line 185 of file ng_l2tp.c.

Referenced by ng_l2tp_rcvmsg(), and ng_l2tp_rcv_lower().

7.118.4.7 [ng_newhook_t](#) [ng_l2tp_newhook](#) [static]

Definition at line 160 of file ng_l2tp.c.

7.118.4.8 [ng_rcvdata_t](#) [ng_l2tp_rcvdata](#) [static]

Definition at line 161 of file ng_l2tp.c.

7.118.4.9 [ng_rcvmsg_t](#) [ng_l2tp_rcvmsg](#) [static]

Definition at line 158 of file ng_l2tp.c.

7.118.4.10 [ng_fn_eachhook](#) [ng_l2tp_reset_session](#) [static]

Definition at line 186 of file ng_l2tp.c.

Referenced by ng_l2tp_seq_reset().

7.118.4.11 `struct ng_parse_struct_field ng_l2tp_seq_config_fields[] =`
`NG_L2TP_SEQ_CONFIG_TYPE_INFO [static]`

Definition at line 194 of file ng_l2tp.c.

7.118.4.12 `struct ng_parse_type ng_l2tp_seq_config_type [static]`**Initial value:**

```
{  
    &ng_parse_struct_type,  
    &ng_l2tp_seq_config_fields  
}
```

Definition at line 195 of file ng_l2tp.c.

7.118.4.13 `struct ng_parse_type ng_l2tp_sess_config_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_l2tp_sess_config_type_fields,
}
```

Definition at line 211 of file ng_l2tp.c.

7.118.4.14 `struct ng_parse_struct_field ng_l2tp_sess_config_type_fields[] = NG_L2TP_SESS_CONFIG_TYPE_INFO` [static]

Definition at line 210 of file ng_l2tp.c.

7.118.4.15 `struct ng_parse_type ng_l2tp_session_stats_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_l2tp_session_stats_type_fields
}
```

Definition at line 227 of file ng_l2tp.c.

7.118.4.16 `struct ng_parse_struct_field ng_l2tp_session_stats_type_fields[] = NG_L2TP_SESSION_STATS_TYPE_INFO` [static]

Definition at line 226 of file ng_l2tp.c.

7.118.4.17 `ng_shutdown_t ng_l2tp_shutdown` [static]

Definition at line 159 of file ng_l2tp.c.

7.118.4.18 `struct ng_parse_type ng_l2tp_stats_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_l2tp_stats_type_fields
}
```

Definition at line 219 of file ng_l2tp.c.

7.118.4.19 `struct ng_parse_struct_field ng_l2tp_stats_type_fields[] = NG_L2TP_STATS_TYPE_INFO` [static]

Definition at line 218 of file ng_l2tp.c.

7.118.4.20 struct `ng_type ng_l2tp_tpestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_L2TP_NODE_TYPE,
    .constructor =  ng_l2tp_constructor,
    .rcvmsg =      ng_l2tp_rcvmsg,
    .shutdown =    ng_l2tp_shutdown,
    .newhook =     ng_l2tp_newhook,
    .rcvdata =     ng_l2tp_rcvdata,
    .disconnect =  ng_l2tp_disconnect,
    .cmdlist =     ng_l2tp_cmdlist,
}
```

Definition at line 322 of file `ng_l2tp.c`.

7.119 /usr/src/sys/netgraph/ng_l2tp.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_l2tp_seq_config](#)
- struct [ng_l2tp_config](#)
- struct [ng_l2tp_sess_config](#)
- struct [ng_l2tp_stats](#)
- struct [ng_l2tp_session_stats](#)

Defines

- #define [NG_L2TP_NODE_TYPE](#) "l2tp"
- #define [NGM_L2TP_COOKIE](#) 1091515793
- #define [NG_L2TP_HOOK_CTRL](#) "ctrl"
- #define [NG_L2TP_HOOK_LOWER](#) "lower"
- #define [NG_L2TP_HOOK_SESSION_P](#) "session_"
- #define [NG_L2TP_HOOK_SESSION_F](#) "session_%04x"
- #define [NG_L2TP_SEQ_CONFIG_TYPE_INFO](#)
- #define [NG_L2TP_CONFIG_TYPE_INFO](#)
- #define [NG_L2TP_SESS_CONFIG_TYPE_INFO](#)
- #define [NG_L2TP_STATS_TYPE_INFO](#)
- #define [NG_L2TP_SESSION_STATS_TYPE_INFO](#)

Enumerations

- enum {
 - [NGM_L2TP_SET_CONFIG](#) = 1, [NGM_L2TP_GET_CONFIG](#), [NGM_L2TP_SET_SESS_CONFIG](#),
 - [NGM_L2TP_GET_SESS_CONFIG](#),
 - [NGM_L2TP_GET_STATS](#), [NGM_L2TP_CLR_STATS](#), [NGM_L2TP_GETCLR_STATS](#), [NGM_L2TP_GET_SESSION_STATS](#),
 - [NGM_L2TP_CLR_SESSION_STATS](#), [NGM_L2TP_GETCLR_SESSION_STATS](#), [NGM_L2TP_ACK_FAILURE](#), [NGM_L2TP_SET_SEQ](#) }

7.119.1 Define Documentation

7.119.1.1 #define [NG_L2TP_CONFIG_TYPE_INFO](#)

Value:

```

{
    { "enabled",          \      &ng_parse_uint8_type   },      \
    { "match_id",       \      &ng_parse_uint8_type   },      \
    { "tunnel_id",     \      &ng_parse_hint16_type  },      \
    { "peer_id",       \      &ng_parse_hint16_type  },      \
    { "peer_win",      \      &ng_parse_uint16_type  },      \
    { "remit_max",     \      &ng_parse_uint16_type  },      \
    { "remit_max_to", \      &ng_parse_uint16_type  },      \
    { NULL }
}

```

Definition at line 84 of file ng_l2tp.h.

7.119.1.2 #define NG_L2TP_HOOK_CTRL "ctrl"

Definition at line 50 of file ng_l2tp.h.

Referenced by ng_l2tp_newhook().

7.119.1.3 #define NG_L2TP_HOOK_LOWER "lower"

Definition at line 51 of file ng_l2tp.h.

Referenced by ng_l2tp_newhook().

7.119.1.4 #define NG_L2TP_HOOK_SESSION_F "session_%04x"

Definition at line 55 of file ng_l2tp.h.

7.119.1.5 #define NG_L2TP_HOOK_SESSION_P "session_"

Definition at line 54 of file ng_l2tp.h.

Referenced by ng_l2tp_newhook().

7.119.1.6 #define NG_L2TP_NODE_TYPE "l2tp"

Definition at line 46 of file ng_l2tp.h.

7.119.1.7 #define NG_L2TP_SEQ_CONFIG_TYPE_INFO

Value:

```

{
    { "ns",          \      &ng_parse_uint16_type  },      \
    { "nr",       \      &ng_parse_uint16_type  },      \
    { NULL }
}

```

Definition at line 66 of file ng_l2tp.h.

7.119.1.8 #define NG_L2TP_SESS_CONFIG_TYPE_INFO**Value:**

```
{
    { "session_id",      &ng_parse_hint16_type }, \
    { "peer_id",        &ng_parse_hint16_type }, \
    { "control_dseq",   &ng_parse_uint8_type  }, \
    { "enable_dseq",    &ng_parse_uint8_type  }, \
    { "include_length", &ng_parse_uint8_type  }, \
    { NULL }
}
```

Definition at line 105 of file ng_l2tp.h.

7.119.1.9 #define NG_L2TP_SESSION_STATS_TYPE_INFO**Value:**

```
{
    { "xmitPackets",    &ng_parse_uint64_type }, \
    { "xmitOctets",     &ng_parse_uint64_type }, \
    { "recvPackets",    &ng_parse_uint64_type }, \
    { "recvOctets",     &ng_parse_uint64_type }, \
    { NULL }
}
```

Definition at line 172 of file ng_l2tp.h.

7.119.1.10 #define NG_L2TP_STATS_TYPE_INFO**Value:**

```
{
    { "xmitPackets",    &ng_parse_uint32_type }, \
    { "xmitOctets",     &ng_parse_uint32_type }, \
    { "xmitZLBs",       &ng_parse_uint32_type }, \
    { "xmitDrops",      &ng_parse_uint32_type }, \
    { "xmitTooBig",     &ng_parse_uint32_type }, \
    { "xmitInvalid",    &ng_parse_uint32_type }, \
    { "xmitDataTooBig", &ng_parse_uint32_type }, \
    { "xmitRetransmits", &ng_parse_uint32_type }, \
    { "recvPackets",    &ng_parse_uint32_type }, \
    { "recvOctets",     &ng_parse_uint32_type }, \
    { "recvRunts",      &ng_parse_uint32_type }, \
    { "recvInvalid",    &ng_parse_uint32_type }, \
    { "recvWrongTunnel", &ng_parse_uint32_type }, \
    { "recvUnknownSID", &ng_parse_uint32_type }, \
    { "recvBadAcks",    &ng_parse_uint32_type }, \
    { "recvOutOfOrder", &ng_parse_uint32_type }, \
    { "recvDuplicates", &ng_parse_uint32_type }, \
    { "recvDataDrops",  &ng_parse_uint32_type }, \
    { "recvZLBs",       &ng_parse_uint32_type }, \
    { "memoryFailures", &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 139 of file ng_l2tp.h.

7.119.1.11 `#define NGM_L2TP_COOKIE 1091515793`

Definition at line 47 of file `ng_l2tp.h`.

Referenced by `ng_l2tp_rcvmsg()`, and `ng_l2tp_seq_failure()`.

7.119.2 Enumeration Type Documentation

7.119.2.1 anonymous enum

Enumerator:

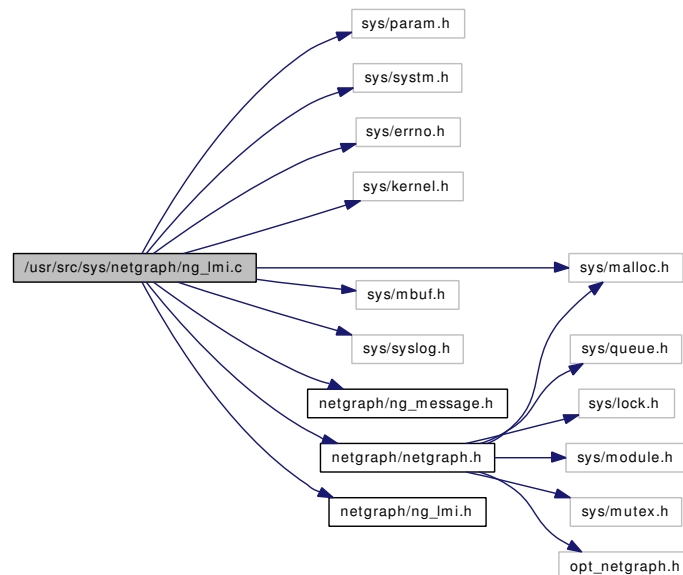
- NGM_L2TP_SET_CONFIG*
- NGM_L2TP_GET_CONFIG*
- NGM_L2TP_SET_SESS_CONFIG*
- NGM_L2TP_GET_SESS_CONFIG*
- NGM_L2TP_GET_STATS*
- NGM_L2TP_CLR_STATS*
- NGM_L2TP_GETCLR_STATS*
- NGM_L2TP_GET_SESSION_STATS*
- NGM_L2TP_CLR_SESSION_STATS*
- NGM_L2TP_GETCLR_SESSION_STATS*
- NGM_L2TP_ACK_FAILURE*
- NGM_L2TP_SET_SEQ*

Definition at line 181 of file `ng_l2tp.h`.

7.120 /usr/src/sys/netgraph/ng_lmi.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/syslog.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_lmi.h>
```

Include dependency graph for ng_lmi.c:



Data Structures

- struct [nglmi_softc](#)

Defines

- #define [NAME_ANNEXA](#) NG_LMI_HOOK_ANNEXA
- #define [NAME_ANNEXD](#) NG_LMI_HOOK_ANNEXD
- #define [NAME_GROUP4](#) NG_LMI_HOOK_GROUPOF4
- #define [NAME_NONE](#) "None"
- #define [MAX_DLCIS](#) 128
- #define [MAXDLCI](#) 1023

- #define `DLCI_NULL` 0
- #define `DLCI_UP` 1
- #define `DLCI_DOWN` 2
- #define `LMI_MIN_LENGTH` 8
- #define `SCF_CONNECTED` 0x01
- #define `SCF_AUTO` 0x02
- #define `SCF_FIXED` 0x04
- #define `SCF_LMITYPE` 0x18
- #define `SCF_NOLMI` 0x00
- #define `SCF_ANNEX_A` 0x08
- #define `SCF_ANNEX_D` 0x10
- #define `SCF_GROUP4` 0x18
- #define `SETLMITYPE`(sc, annex)
- #define `NOPROTO`(sc) (((sc) → flags & SCF_LMITYPE) == SCF_NOLMI)
- #define `ANNEXA`(sc) (((sc) → flags & SCF_LMITYPE) == SCF_ANNEX_A)
- #define `ANNEXD`(sc) (((sc) → flags & SCF_LMITYPE) == SCF_ANNEX_D)
- #define `GROUP4`(sc) (((sc) → flags & SCF_LMITYPE) == SCF_GROUP4)
- #define `LMIPOLLSIZE` 3
- #define `LMI_PATIENCE` 8
- #define `STEPBY`(stepsize)

Typedefs

- typedef `nglmi_softc` * `sc_p`

Functions

- static int `nglmi_checkdata` (`hook_p` hook, struct mbuf *m)
- `NETGRAPH_INIT` (lmi,&typestruct)
- static void `LMI_ticker` (`node_p` node, `hook_p` hook, void *arg1, int arg2)
- static void `nglmi_startup_fixed` (`sc_p` sc, `hook_p` hook)
- static void `nglmi_startup_auto` (`sc_p` sc)
- static void `nglmi_startup` (`sc_p` sc)
- static void `nglmi_inquire` (`sc_p` sc, int full)
- static void `ngauto_state_machine` (`sc_p` sc)
- static int `nglmi_constructor` (`node_p` node)
- static int `nglmi_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `nglmi_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `nglmi_rcvdata` (`hook_p` hook, `item_p` item)
- static int `nglmi_shutdown` (`node_p` node)
- static int `nglmi_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `nglmi_constructor`
- static `ng_rcvmsg_t` `nglmi_rcvmsg`
- static `ng_shutdown_t` `nglmi_shutdown`
- static `ng_newhook_t` `nglmi_newhook`
- static `ng_rcvdata_t` `nglmi_rcvdata`
- static `ng_disconnect_t` `nglmi_disconnect`
- static struct `ng_type` `typestruct`

7.120.1 Define Documentation

7.120.1.1 **#define ANNEXA(sc) (((sc) → flags & SCF_LMITYPE) == SCF_ANNEX_A)**

Definition at line 173 of file ng_lmi.c.

Referenced by nglmi_inquire(), and nglmi_rcvdata().

7.120.1.2 **#define ANNEXD(sc) (((sc) → flags & SCF_LMITYPE) == SCF_ANNEX_D)**

Definition at line 174 of file ng_lmi.c.

Referenced by nglmi_checkdata(), nglmi_inquire(), and nglmi_rcvdata().

7.120.1.3 **#define DLCI_DOWN 2**

Definition at line 83 of file ng_lmi.c.

Referenced by nglmi_inquire(), nglmi_rcvdata(), and nglmi_rcvmsg().

7.120.1.4 **#define DLCI_NULL 0**

Definition at line 81 of file ng_lmi.c.

7.120.1.5 **#define DLCI_UP 1**

Definition at line 82 of file ng_lmi.c.

Referenced by nglmi_inquire(), nglmi_rcvdata(), and nglmi_rcvmsg().

7.120.1.6 **#define GROUP4(sc) (((sc) → flags & SCF_LMITYPE) == SCF_GROUP4)**

Definition at line 175 of file ng_lmi.c.

Referenced by nglmi_checkdata(), nglmi_inquire(), and nglmi_rcvdata().

7.120.1.7 **#define LMI_MIN_LENGTH 8**

Definition at line 88 of file ng_lmi.c.

7.120.1.8 **#define LMI_PATIENCE 8**

Definition at line 178 of file ng_lmi.c.

Referenced by nglmi_inquire().

7.120.1.9 **#define LMIPOLLSIZE 3**

Definition at line 177 of file ng_lmi.c.

Referenced by ngauto_state_machine().

7.120.1.10 #define MAX_DLCIS 128

Definition at line 75 of file ng_lmi.c.

7.120.1.11 #define MAXDLCI 1023

Definition at line 76 of file ng_lmi.c.

Referenced by nglmi_inquire(), nglmi_rcvdata(), and nglmi_rcvmsg().

7.120.1.12 #define NAME_ANNEXA NG_LMI_HOOK_ANNEXA

Definition at line 70 of file ng_lmi.c.

Referenced by nglmi_newhook(), and nglmi_rcvdata().

7.120.1.13 #define NAME_ANNEXD NG_LMI_HOOK_ANNEXD

Definition at line 71 of file ng_lmi.c.

Referenced by nglmi_newhook(), and nglmi_rcvdata().

7.120.1.14 #define NAME_GROUP4 NG_LMI_HOOK_GROUPOF4

Definition at line 72 of file ng_lmi.c.

Referenced by nglmi_newhook(), and nglmi_rcvdata().

7.120.1.15 #define NAME_NONE "None"

Definition at line 73 of file ng_lmi.c.

Referenced by nglmi_constructor(), and nglmi_newhook().

7.120.1.16 #define NOPROTO(sc) (((sc) → flags & SCF_LMITYPE) == SCF_NOLMI)

Definition at line 172 of file ng_lmi.c.

7.120.1.17 #define SCF_ANNEX_A 0x08

Definition at line 162 of file ng_lmi.c.

Referenced by ngauto_state_machine(), nglmi_checkdata(), and nglmi_newhook().

7.120.1.18 #define SCF_ANNEX_D 0x10

Definition at line 163 of file ng_lmi.c.

Referenced by ngauto_state_machine(), nglmi_checkdata(), and nglmi_newhook().

7.120.1.19 #define SCF_AUTO 0x02

Definition at line 157 of file ng_lmi.c.

Referenced by LMI_ticker(), nglmi_checkdata(), nglmi_rcvdata(), nglmi_rcvmsg(), and nglmi_startup_auto().

7.120.1.20 #define SCF_CONNECTED 0x01

Definition at line 156 of file ng_lmi.c.

Referenced by nglmi_disconnect(), nglmi_newhook(), nglmi_startup_auto(), and nglmi_startup_fixed().

7.120.1.21 #define SCF_FIXED 0x04

Definition at line 158 of file ng_lmi.c.

Referenced by nglmi_rcvmsg(), and nglmi_startup_fixed().

7.120.1.22 #define SCF_GROUP4 0x18

Definition at line 164 of file ng_lmi.c.

Referenced by ngauto_state_machine(), nglmi_checkdata(), and nglmi_newhook().

7.120.1.23 #define SCF_LMITYPE 0x18

Definition at line 160 of file ng_lmi.c.

7.120.1.24 #define SCF_NOLMI 0x00

Definition at line 161 of file ng_lmi.c.

Referenced by nglmi_checkdata().

7.120.1.25 #define SETLMITYPE(sc, annex)**Value:**

```
do {
    (sc)->flags &= ~SCF_LMITYPE;
    (sc)->flags |= (annex);
} while (0)
```

Definition at line 166 of file ng_lmi.c.

Referenced by ngauto_state_machine(), nglmi_checkdata(), and nglmi_newhook().

7.120.1.26 #define STEPBY(stepsize)**Value:**

```
do {
    packetlen -= (stepsize); \
    data += (stepsize); \
} while (0)
```

Definition at line 544 of file ng_lmi.c.

Referenced by nglmi_checkdata(), and nglmi_rcvdata().

7.120.2 Typedef Documentation

7.120.2.1 typedef struct nglmi_softc* sc_p

Definition at line 140 of file ng_lmi.c.

7.120.3 Function Documentation

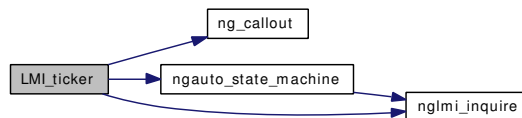
7.120.3.1 static void LMI_ticker (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 264 of file ng_lmi.c.

References ng_callout(), NG_LMI_POLL_RATE, NG_NODE_PRIVATE, ngauto_state_machine(), nglmi_inquire(), nglmi_softc::node, and SCF_AUTO.

Referenced by nglmi_startup().

Here is the call graph for this function:



7.120.3.2 NETGRAPH_INIT (lmi, & typestruct)

7.120.3.3 static void ngauto_state_machine (sc_p sc) [static]

Definition at line 395 of file ng_lmi.c.

References LMIPOLLSIZE, nglmi_inquire(), SCF_ANNEX_A, SCF_ANNEX_D, SCF_GROUP4, and SETLMITYPE.

Referenced by LMI_ticker().

Here is the call graph for this function:



7.120.3.4 static int nglmi_checkdata (hook_p hook, struct mbuf * m) [static]

Definition at line 737 of file ng_lmi.c.

References ANNEXD, GROUP4, NG_FREE_M, NG_HOOK_NODE, NG_NODE_PRIVATE, SCF_ANNEX_A, SCF_ANNEX_D, SCF_AUTO, SCF_GROUP4, SCF_NOLMI, SETLMITYPE, and STEPBY.

Referenced by nglmi_rcvdata().

7.120.3.5 static int nglmi_constructor (node_p node) [static]

Definition at line 184 of file ng_lmi.c.

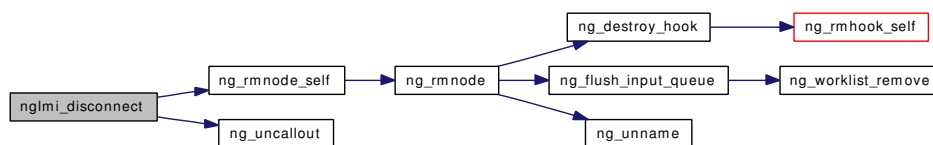
References NAME_NONE, ng_callout_init, NG_LMI_KEEPALIVE_RATE, NG_LMI_SEQ_PER_FULL, NG_NODE_SET_PRIVATE, and nglmi_softc::node.

7.120.3.6 static int nglmi_disconnect (hook_p hook) [static]

Definition at line 1065 of file ng_lmi.c.

References NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_IS_VALID, NG_NODE_PRIVATE, ng_rmnode_self(), ng_uncallout(), ng_async_private::node, and SCF_CONNECTED.

Here is the call graph for this function:

**7.120.3.7 static void nglmi_inquire (sc_p sc, int full) [static]**

Definition at line 313 of file ng_lmi.c.

References ANNEXA, ANNEXD, ng_tag_prio::discardability, DLCI_DOWN, DLCI_UP, GROUP4, LMI_PATIENCE, MAXDLCI, NG_PRIO_LINKSTATE, NG_SEND_DATA_ONLY, NG_TAG_PRIO, NGM_GENERIC_COOKIE, ng_tag_prio::priority, and ng_tag_prio::tag.

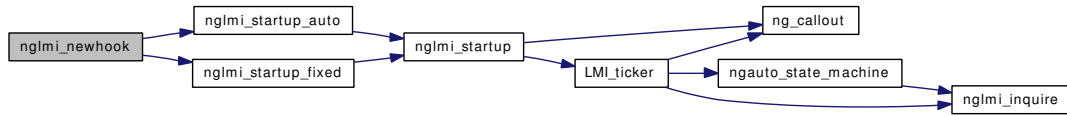
Referenced by LMI_ticker(), and ngauto_state_machine().

7.120.3.8 static int nglmi_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 207 of file ng_lmi.c.

References NAME_ANNEXA, NAME_ANNEXD, NAME_GROUP4, NAME_NONE, NG_HOOK_SET_PRIVATE, NG_LMI_HOOK_ANNEXA, NG_LMI_HOOK_ANNEXD, NG_LMI_HOOK_AUTO0, NG_LMI_HOOK_AUTO1023, NG_LMI_HOOK_DEBUG, NG_LMI_HOOK_GROUPOF4, NG_NODE_PRIVATE, nglmi_startup_auto(), nglmi_startup_fixed(), nglmi_softc::node, SCF_ANNEX_A, SCF_ANNEX_D, SCF_CONNECTED, SCF_GROUP4, and SETLMITYPE.

Here is the call graph for this function:



7.120.3.9 static int nglmi_rcvdata (*hook_p* hook, *item_p* item) [static]

Definition at line 555 of file ng_lmi.c.

References ANNEXA, ANNEXD, DLCI_DOWN, DLCI_UP, GROUP4, nglmistat::hook, MAXDLCI, NAME_ANNEXA, NAME_ANNEXD, NAME_GROUP4, NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, NGI_GET_M, nglmi_checkdata(), SCF_AUTO, and STEPBY.

Here is the call graph for this function:



7.120.3.10 static int nglmi_rcvmsg (*node_p* node, *item_p* item, *hook_p* lasthook) [static]

Definition at line 443 of file ng_lmi.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, DLCI_DOWN, DLCI_UP, ng_mesg::header, MAXDLCI, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_TEXTRESPONSE, NGI_GET_MSG, NGM_GENERIC_COOKIE, NGM_LMI_COOKIE, NGM_LMI_GET_STATUS, NGM_TEXT_STATUS, SCF_AUTO, SCF_FIXED, and ng_mesg::ng_msghdr::typecookie.

7.120.3.11 static int nglmi_shutdown (*node_p* node) [static]

Definition at line 1050 of file ng_lmi.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_async_private::node.

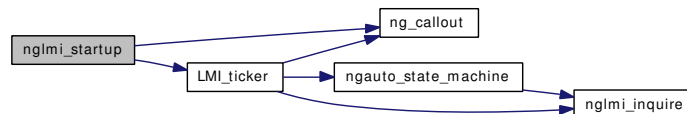
7.120.3.12 static void nglmi_startup (*sc_p* sc) [static]

Definition at line 302 of file ng_lmi.c.

References LMI_ticker(), ng_callout(), and ng_async_private::node.

Referenced by nglmi_startup_auto(), and nglmi_startup_fixed().

Here is the call graph for this function:



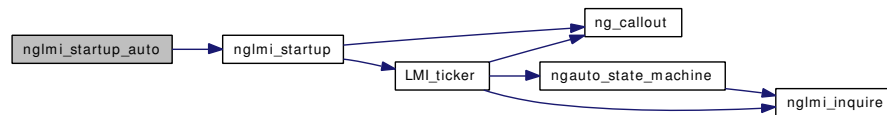
7.120.3.13 static void nglmi_startup_auto (sc_p sc) [static]

Definition at line 293 of file ng_lmi.c.

References nglmi_startup(), SCF_AUTO, and SCF_CONNECTED.

Referenced by nglmi_newhook().

Here is the call graph for this function:

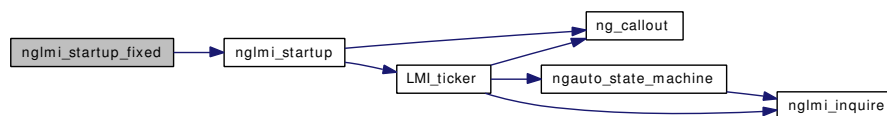
**7.120.3.14 static void nglmi_startup_fixed (sc_p sc, hook_p hook) [static]**

Definition at line 285 of file ng_lmi.c.

References nglmi_startup(), SCF_CONNECTED, and SCF_FIXED.

Referenced by nglmi_newhook().

Here is the call graph for this function:

**7.120.4 Variable Documentation****7.120.4.1 ng_constructor_t nglmi_constructor [static]**

Definition at line 93 of file ng_lmi.c.

7.120.4.2 ng_disconnect_t nglmi_disconnect [static]

Definition at line 98 of file ng_lmi.c.

7.120.4.3 ng_newhook_t nglmi_newhook [static]

Definition at line 96 of file ng_lmi.c.

7.120.4.4 ng_rcvdata_t nglmi_rcvdata [static]

Definition at line 97 of file ng_lmi.c.

7.120.4.5 `ng_rcvmsg_t nglmi_rcvmsg` [static]

Definition at line 94 of file ng_lmi.c.

7.120.4.6 `ng_shutdown_t nglmi_shutdown` [static]

Definition at line 95 of file ng_lmi.c.

7.120.4.7 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_LMI_NODE_TYPE,
    .constructor =  nglmi_constructor,
    .rcvmsg =      nglmi_rcvmsg,
    .shutdown =    nglmi_shutdown,
    .newhook =     nglmi_newhook,
    .rcvdata =     nglmi_rcvdata,
    .disconnect =  nglmi_disconnect,
}
```

Definition at line 101 of file ng_lmi.c.

7.121 /usr/src/sys/netgraph/ng_lmi.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [nglmistat](#)

Defines

- #define [NG_LMI_NODE_TYPE](#) "lmi"
- #define [NGM_LMI_COOKIE](#) 867184133
- #define [NG_LMI_HOOK_DEBUG](#) "debug"
- #define [NG_LMI_HOOK_ANNEXA](#) "annexA"
- #define [NG_LMI_HOOK_ANNEXD](#) "annexD"
- #define [NG_LMI_HOOK_GROUPOF4](#) "group4"
- #define [NG_LMI_HOOK_AUTO0](#) "auto0"
- #define [NG_LMI_HOOK_AUTO1023](#) "auto1023"
- #define [NGM_LMI_STAT_ARYSIZE](#) (1024/8)
- #define [NG_LMI_KEEPA_LIVE_RATE](#) 10
- #define [NG_LMI_POLL_RATE](#) 3
- #define [NG_LMI_SEQ_PER_FULL](#) 5
- #define [NG_LMI_LMI_PRIORITY](#) 64

Enumerations

- enum { [NGM_LMI_GET_STATUS](#) = 1 }

7.121.1 Define Documentation

7.121.1.1 #define [NG_LMI_HOOK_ANNEXA](#) "annexA"

Definition at line 53 of file [ng_lmi.h](#).

Referenced by [nglmi_newhook\(\)](#).

7.121.1.2 #define [NG_LMI_HOOK_ANNEXD](#) "annexD"

Definition at line 54 of file [ng_lmi.h](#).

Referenced by [nglmi_newhook\(\)](#).

7.121.1.3 #define [NG_LMI_HOOK_AUTO0](#) "auto0"

Definition at line 56 of file [ng_lmi.h](#).

Referenced by [nglmi_newhook\(\)](#).

7.121.1.4 #define NG_LMI_HOOK_AUTO1023 "auto1023"

Definition at line 57 of file ng_lmi.h.

Referenced by nglmi_newhook().

7.121.1.5 #define NG_LMI_HOOK_DEBUG "debug"

Definition at line 52 of file ng_lmi.h.

Referenced by nglmi_newhook().

7.121.1.6 #define NG_LMI_HOOK_GROUPOF4 "group4"

Definition at line 55 of file ng_lmi.h.

Referenced by nglmi_newhook().

7.121.1.7 #define NG_LMI_KEEPLIVE_RATE 10

Definition at line 76 of file ng_lmi.h.

Referenced by nglmi_constructor().

7.121.1.8 #define NG_LMI_LMI_PRIORITY 64

Definition at line 79 of file ng_lmi.h.

7.121.1.9 #define NG_LMI_NODE_TYPE "lmi"

Definition at line 48 of file ng_lmi.h.

7.121.1.10 #define NG_LMI_POLL_RATE 3

Definition at line 77 of file ng_lmi.h.

Referenced by LMI_ticker().

7.121.1.11 #define NG_LMI_SEQ_PER_FULL 5

Definition at line 78 of file ng_lmi.h.

Referenced by nglmi_constructor().

7.121.1.12 #define NGM_LMI_COOKIE 867184133

Definition at line 49 of file ng_lmi.h.

Referenced by nglmi_rcvmsg().

7.121.1.13 #define NGM_LMI_STAT_ARYSIZE (1024/8)

Definition at line 64 of file ng_lmi.h.

7.121.2 Enumeration Type Documentation

7.121.2.1 anonymous enum

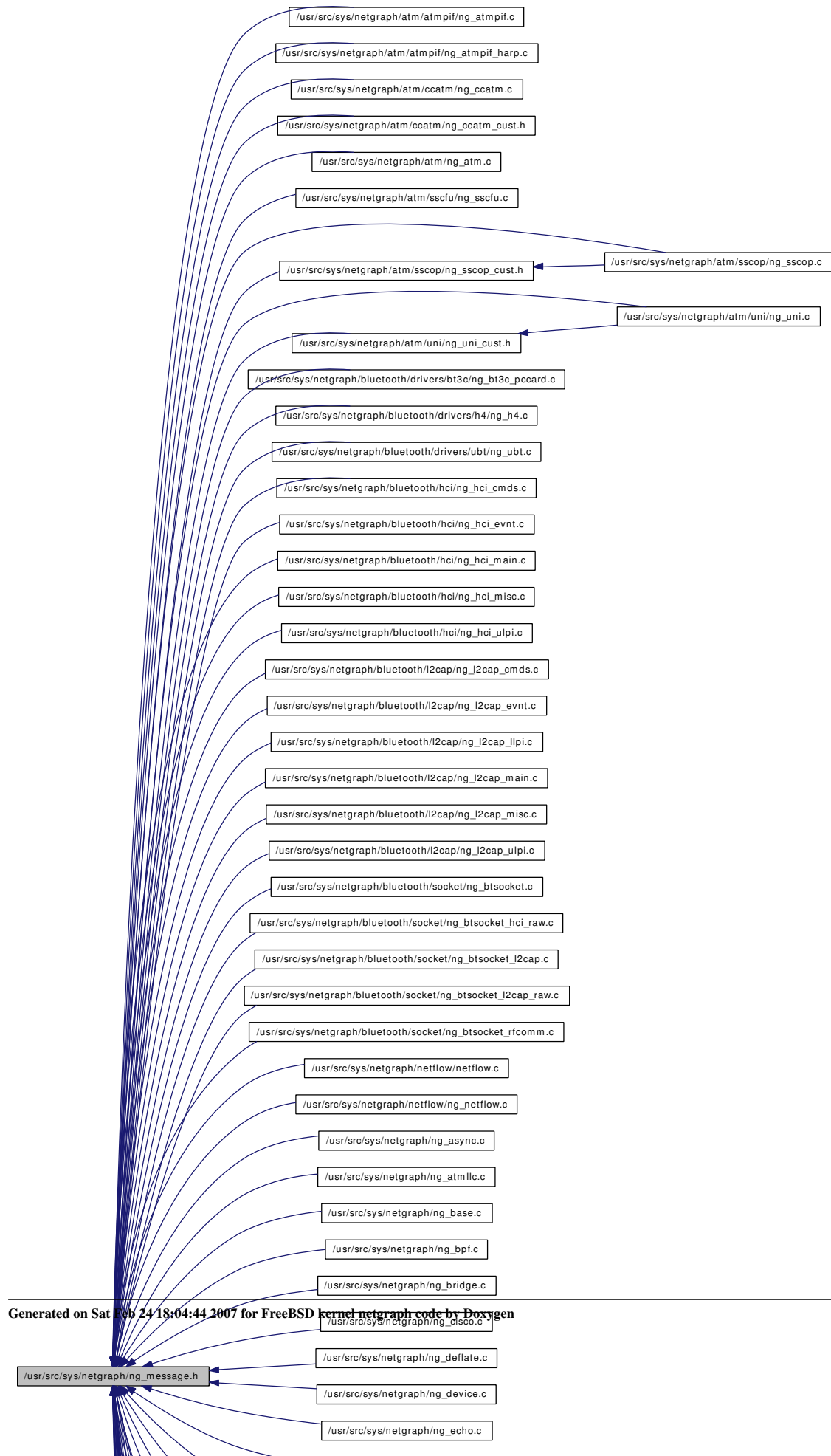
Enumerator:

NGM_LMI_GET_STATUS

Definition at line 60 of file ng_lmi.h.

7.122 /usr/src/sys/netgraph/ng_message.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_mesg](#)
- struct [ng_mesg::ng_msghdr](#)
- struct [ngm_mkpeer](#)
- struct [ngm_connect](#)
- struct [ngm_name](#)
- struct [ngm_rmhook](#)
- struct [nodeinfo](#)
- struct [linkinfo](#)
- struct [hooklist](#)
- struct [namelist](#)
- struct [typeinfo](#)
- struct [typelist](#)
- struct [ngm_bandwidth](#)
- struct [ngm_queue_state](#)
- struct [flow_manager](#)

Defines

- #define [NG_TYPESIZ](#) 32
- #define [NG_HOOKSIZ](#) 32
- #define [NG_NODESIZ](#) 32
- #define [NG_PATHSIZ](#) 512
- #define [NG_CMDSTRSIZ](#) 32
- #define [NG_TYPELEN](#) (NG_TYPESIZ - 1)
- #define [NG_HOOKLEN](#) (NG_HOOKSIZ - 1)
- #define [NG_NODELEN](#) (NG_NODESIZ - 1)
- #define [NG_PATHLEN](#) (NG_PATHSIZ - 1)
- #define [NG_CMDSTRLEN](#) (NG_CMDSTRSIZ - 1)
- #define [NG_TEXTRESPONSE](#) 1024
- #define [NGM_READONLY](#) 0x10000000
- #define [NGM_HASREPLY](#) 0x20000000
- #define [NG_GENERIC_NG_MESG_INFO](#)(dtype)
- #define [NG_VERSION](#) 8
- #define [NGF_ORIG](#) 0x00000000
- #define [NGF_RESP](#) 0x00000001
- #define [ng_ID_t](#) uint32_t
- #define [NGM_GENERIC_COOKIE](#) 1137070366
- #define [NGM_FLOW_COOKIE](#) 851672669
- #define [NGM_LINK_IS_UP](#) 32
- #define [NGM_LINK_IS_DOWN](#) 33
- #define [NGM_HIGH_WATER_PASSED](#) 34
- #define [NGM_LOW_WATER_PASSED](#) 35
- #define [NGM_SYNC_QUEUE_STATE](#) 36
- #define [NGM_DROP_LINK](#) 41
- #define [NGM_RAISE_LINK](#) 42
- #define [NGM_FLUSH_QUEUE](#) 43
- #define [NGM_GET_BANDWIDTH](#) (44|NGM_READONLY)
- #define [NGM_SET_XMIT_Q_LIMITS](#) 45

- #define `NGM_GET_XMIT_Q_LIMITS` (46|NGM_READONLY)
- #define `NGM_MICROMANAGE` 47
- #define `NGM_SET_FLOW_MANAGER` 48
- #define `NG_GENERIC_MKPEER_INFO`()
- #define `NG_GENERIC_CONNECT_INFO`()
- #define `NG_GENERIC_NAME_INFO`()
- #define `NG_GENERIC_RMHOOK_INFO`()
- #define `NG_GENERIC_NODEINFO_INFO`()
- #define `NG_GENERIC_LINKINFO_INFO`(nitype)
- #define `NG_GENERIC_HOOKLIST_INFO`(nitype, litype)
- #define `NG_GENERIC_LISTNODES_INFO`(niarraytype)
- #define `NG_GENERIC_TYPEINFO_INFO`()
- #define `NG_GENERIC_TYPELIST_INFO`(tiarraytype)
- #define `NG_GENERIC_BANDWIDTH_INFO`()
- #define `NG_GENERIC_QUEUE_INFO`()
- #define `NG_GENERIC_FLOW_MANAGER_INFO`()
- #define `NGIOCGINFO_IOR`('N', 40, struct `nodeinfo`)
- #define `NGIOCSETNAME_IOW`('N', 41, struct `ngm_name`)
- #define `NG_MKMESSAGE`(msg, cookie, cmdid, len, how)
- #define `NG_MKRESPONSE`(rsp, msg, len, how)
- #define `NG_COPYMESSAGE`(copy, msg, how)

Enumerations

- enum {
 - `NGM_SHUTDOWN` = 1, `NGM_MKPEER` = 2, `NGM_CONNECT` = 3, `NGM_NAME` = 4,
 - `NGM_RMHOOK` = 5, `NGM_NODEINFO` = (6|NGM_READONLY|NGM_HASREPLY), `NGM_LISTHOOKS` = (7|NGM_READONLY|NGM_HASREPLY), `NGM_LISTNAMES` = (8|NGM_READONLY|NGM_HASREPLY),
 - `NGM_LISTNODES` = (9|NGM_READONLY|NGM_HASREPLY), `NGM_LISTTYPES` = (10|NGM_READONLY|NGM_HASREPLY), `NGM_TEXT_STATUS` = (11|NGM_READONLY|NGM_HASREPLY), `NGM_BINARY2ASCII` = (12|NGM_READONLY|NGM_HASREPLY),
 - `NGM_ASCII2BINARY` = (13|NGM_READONLY|NGM_HASREPLY), `NGM_TEXT_CONFIG` = 14 }

7.122.1 Define Documentation

7.122.1.1 #define `NG_CMDSTRLEN` (`NG_CMDSTRSIZ` - 1)

Definition at line 60 of file `ng_message.h`.

7.122.1.2 #define `NG_CMDSTRSIZ` 32

Definition at line 52 of file `ng_message.h`.

7.122.1.3 #define NG_COPYMESSAGE(copy, msg, how)**Value:**

```

do {
    MALLOC((copy), struct ng_mesg *, sizeof(struct ng_mesg) + \
        (msg->header.arglen, M_NETGRAPH_MSG, (how) | M_ZERO); \
    if ((copy) == NULL) \
        break; \
    (copy)->header.version = NG_VERSION; \
    (copy)->header.arglen = (msg)->header.arglen; \
    (copy)->header.token = (msg)->header.token; \
    (copy)->header.typecookie = (msg)->header.typecookie; \
    (copy)->header.cmd = (msg)->header.cmd; \
    (copy)->header.flags = (msg)->header.flags; \
    bcopy((msg)->header.cmdstr, (copy)->header.cmdstr, \
        sizeof((copy)->header.cmdstr)); \
    if ((msg)->header.arglen > 0) \
        bcopy((msg)->data, (copy)->data, (msg)->header.arglen); \
} while (0)

```

Definition at line 426 of file ng_message.h.

Referenced by ng_vlan_rcvmsg().

7.122.1.4 #define NG_GENERIC_BANDWIDTH_INFO()**Value:**

```

{
    { "nominal_in",      &ng_parse_uint64_type }, \
    { "seen_in",        &ng_parse_uint64_type }, \
    { "nominal_out",    &ng_parse_uint64_type }, \
    { "seen_out",       &ng_parse_uint64_type }, \
    { NULL }
}

```

Definition at line 322 of file ng_message.h.

7.122.1.5 #define NG_GENERIC_CONNECT_INFO()**Value:**

```

{
    { "path",           &ng_parse_pathbuf_type }, \
    { "ourhook",        &ng_parse_hookbuf_type }, \
    { "peerhook",       &ng_parse_hookbuf_type }, \
    { NULL }
}

```

Definition at line 203 of file ng_message.h.

7.122.1.6 #define NG_GENERIC_FLOW_MANAGER_INFO()**Value:**

```
{
    { "id",          \          &ng_parse_hint32_type  }, \
    { NULL }
}
```

Definition at line 365 of file ng_message.h.

7.122.1.7 #define NG_GENERIC_HOOKLIST_INFO(nitype, litype)

Value:

```
{
    { "nodeinfo",    (nitype)          }, \
    { "linkinfo",    (litype)          }, \
    { NULL }
}
```

Definition at line 270 of file ng_message.h.

7.122.1.8 #define NG_GENERIC_LINKINFO_INFO(nitype)

Value:

```
{
    { "ourhook",     &ng_parse_hookbuf_type }, \
    { "peerhook",   &ng_parse_hookbuf_type }, \
    { "nodeinfo",   (nitype)          }, \
    { NULL }
}
```

Definition at line 257 of file ng_message.h.

7.122.1.9 #define NG_GENERIC_LISTNODES_INFO(niarraytype)

Value:

```
{
    { "numnames",    &ng_parse_uint32_type  }, \
    { "nodeinfo",    (niarraytype)        }, \
    { NULL }
}
```

Definition at line 283 of file ng_message.h.

7.122.1.10 #define NG_GENERIC_MKPEER_INFO()

Value:

```
{
    { "type",        \          &ng_parse_typebuf_type  }, \
    { "ourhook",     &ng_parse_hookbuf_type  }, \
    { "peerhook",   &ng_parse_hookbuf_type  }, \
    { NULL }
}
```

Definition at line 188 of file ng_message.h.

7.122.1.11 #define NG_GENERIC_NAME_INFO()**Value:**

```
{
    { "name",          \
      &ng_parse_nodebuf_type }, \
    { NULL }
}
```

Definition at line 216 of file ng_message.h.

7.122.1.12 #define NG_GENERIC_NG_MESG_INFO(dtype)**Value:**

```
{
    { "version",      \
      &ng_parse_uint8_type }, \
    { "spare",       \
      &ng_parse_uint8_type }, \
    { "spare2",      \
      &ng_parse_uint16_type }, \
    { "arglen",      \
      &ng_parse_uint32_type }, \
    { "cmd",         \
      &ng_parse_uint32_type }, \
    { "flags",       \
      &ng_parse_hint32_type }, \
    { "token",       \
      &ng_parse_uint32_type }, \
    { "typecookie", \
      &ng_parse_uint32_type }, \
    { "cmdstr",      \
      &ng_parse_cmdbuf_type }, \
    { "data",        \
      (dtype) }, \
    { NULL }
}
```

Definition at line 87 of file ng_message.h.

7.122.1.13 #define NG_GENERIC_NODEINFO_INFO()**Value:**

```
{
    { "name",          \
      &ng_parse_nodebuf_type }, \
    { "type",         \
      &ng_parse_typebuf_type }, \
    { "id",           \
      &ng_parse_hint32_type }, \
    { "hooks",        \
      &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 241 of file ng_message.h.

7.122.1.14 #define NG_GENERIC_QUEUE_INFO()**Value:**

```
{
    { "max_queuelen_bytes", \
      &ng_parse_uint_type }, \
    { "max_queuelen_packets", \
      &ng_parse_uint_type }, \
    { "high_watermark",    \
      &ng_parse_uint_type }, \
    { "low_watermark",     \
      &ng_parse_uint_type }, \
    { "current",           \
      &ng_parse_uint_type }, \
    { NULL }
}
```

Definition at line 350 of file ng_message.h.

7.122.1.15 #define NG_GENERIC_RMHOOK_INFO()**Value:**

```
{
    { "hook",          \
      { NULL }        \
    }
}
```

Definition at line 227 of file ng_message.h.

7.122.1.16 #define NG_GENERIC_TYPEINFO_INFO()**Value:**

```
{
    { "typename",     \
      { "numnodes",  \
        { NULL }     \
      }
    }
}
```

Definition at line 296 of file ng_message.h.

7.122.1.17 #define NG_GENERIC_TYPELIST_INFO(tiarraytype)**Value:**

```
{
    { "numtypes",    \
      { "typeinfo", \
        { NULL }    \
      }
    }
}
```

Definition at line 308 of file ng_message.h.

7.122.1.18 #define NG_HOOKLEN (NG_HOOKSIZ - 1)

Definition at line 57 of file ng_message.h.

7.122.1.19 #define NG_HOOKSIZ 32

Definition at line 49 of file ng_message.h.

Referenced by ng_add_hook(), ng_atm_rcvmsg(), ng_con_nodes(), ng_ksocket_newhook(), ng_tcpmss_rcvmsg(), ng_vlan_rcvmsg(), ngd_send(), ngh_rcvmsg(), ngs_rcvdata(), pppoe_send_event(), and send_acname().

7.122.1.20 #define ng_ID_t uint32_t

Definition at line 113 of file ng_message.h.

Referenced by ng_decodeidname(), ng_ksocket_incoming2(), ng_ksocket_rcvmsg(), ng_name2noderef(), and ngs_rcvmsg().

7.122.1.21 #define NG_MKMESSAGE(msg, cookie, cmdid, len, how)**Value:**

```
do {
    MALLOC((msg), struct ng_mesg *, sizeof(struct ng_mesg) \
          + (len), M_NETGRAPH_MSG, (how) | M_ZERO); \
    if ((msg) == NULL) \
        break; \
    (msg)->header.version = NG_VERSION; \
    (msg)->header.typecookie = (cookie); \
    (msg)->header.cmd = (cmdid); \
    (msg)->header.arglen = (len); \
    strncpy((msg)->header.cmdstr, #cmdid, \
            sizeof((msg)->header.cmdstr) - 1); \
} while (0)
```

Definition at line 389 of file ng_message.h.

Referenced by cisco_input(), cisco_notify(), ng_atm_event_func(), ng_btsocket_hci_raw_control(), ng_btsocket_hci_raw_send_ngmsg(), ng_btsocket_hci_raw_send_sync_ngmsg(), ng_btsocket_l2cap_raw_control(), ng_btsocket_l2cap_raw_send_ngmsg(), ng_btsocket_l2cap_raw_send_sync_ngmsg(), ng_btsocket_l2cap_send_l2ca_cfg_req(), ng_btsocket_l2cap_send_l2ca_cfg_rsp(), ng_btsocket_l2cap_send_l2ca_con_req(), ng_btsocket_l2cap_send_l2ca_con_rsp_req(), ng_btsocket_l2cap_send_l2ca_discon_req(), ng_deflate_rcvdata(), ng_ether_link_state(), ng_hci_lp_acl_con_req(), ng_hci_lp_con_cfm(), ng_hci_lp_con_ind(), ng_hci_lp_discon_ind(), ng_hci_lp_qos_cfm(), ng_hci_lp_qos_ind(), ng_hci_node_is_up(), ng_ksocket_finish_accept(), ng_ksocket_incoming2(), ng_l2cap_l2ca_cfg_ind(), ng_l2cap_l2ca_cfg_rsp(), ng_l2cap_l2ca_cfg_rsp_rsp(), ng_l2cap_l2ca_con_ind(), ng_l2cap_l2ca_con_rsp(), ng_l2cap_l2ca_con_rsp_rsp(), ng_l2cap_l2ca_discon_ind(), ng_l2cap_l2ca_discon_rsp(), ng_l2cap_l2ca_enable_clt(), ng_l2cap_l2ca_get_info_rsp(), ng_l2cap_l2ca_ping_rsp(), ng_l2cap_l2ca_qos_ind(), ng_l2cap_l2ca_write_rsp(), ng_l2cap_lp_con_ind(), ng_l2cap_lp_con_req(), ng_l2cap_lp_qos_req(), ng_l2cap_process_discon_timeout(), ng_l2cap_send_hook_info(), ng_l2tp_seq_failure(), ng_mppc_rcvdata(), ng_one2many_notify(), ng_pppoe_connect(), ng_pred1_rcvdata(), ng_source_connect(), ng_source_set_autosrc(), pppoe_send_event(), send_acname(), send_sessionid(), and sync_con_queue().

7.122.1.22 #define NG_MKRESPONSE(rsp, msg, len, how)**Value:**

```
do {
    MALLOC((rsp), struct ng_mesg *, sizeof(struct ng_mesg) \
          + (len), M_NETGRAPH_MSG, (how) | M_ZERO); \
    if ((rsp) == NULL) \
        break; \
    (rsp)->header.version = NG_VERSION; \
    (rsp)->header.arglen = (len); \
    (rsp)->header.token = (msg)->header.token; \
    (rsp)->header.typecookie = (msg)->header.typecookie; \
    (rsp)->header.cmd = (msg)->header.cmd; \
    bcopy((msg)->header.cmdstr, (rsp)->header.cmdstr, \
          sizeof((rsp)->header.cmdstr)); \
    (rsp)->header.flags |= NGF_RESP; \
} while (0)
```

Definition at line 407 of file ng_message.h.

Referenced by cisco_rcvmsg(), ng_atm_rcvmsg(), ng_atmpif_rcvmsg(), ng_bpf_rcvmsg(), ng_bridge_rcvmsg(), ng_bt3c_rcvmsg(), ng_ccatm_get_addresses(), ng_ccatm_rcvmsg(), ng_deflate_rcvmsg(), ng_device_rcvmsg(), ng_eiface_rcvmsg(), ng_etc_rcvmsg(), ng_ether_rcvmsg(), ng_generic_msg(), ng_gif_

rcvmsg(), ng_h4_rcvmsg(), ng_hci_default_rcvmsg(), ng_iface_rcvmsg(), ng_ksocket_rcvmsg(), ng_l2cap_default_rcvmsg(), ng_l2tp_rcvmsg(), ng_netflow_rcvmsg(), ng_one2many_rcvmsg(), ng_ppp_rcvmsg(), ng_pppoe_rcvmsg(), ng_pptpgre_rcvmsg(), ng_pred1_rcvmsg(), ng_rfc1490_rcvmsg(), ng_source_rcvmsg(), ng_sppp_rcvmsg(), ng_sscfu_rcvmsg(), ng_sscop_rcvmsg(), ng_tag_rcvmsg(), ng_tpmss_rcvmsg(), ng_ibt_rcvmsg(), ng_uni_rcvmsg(), ng_vjc_rcvmsg(), ng_vlan_rcvmsg(), ng_xxx_rcvmsg(), nga_rcvmsg(), ngh_rcvmsg(), nglmi_rcvmsg(), and ngt_rcvmsg().

7.122.1.23 #define NG_NODELEN (NG_NODESIZ - 1)

Definition at line 58 of file ng_message.h.

7.122.1.24 #define NG_NODESIZ 32

Definition at line 50 of file ng_message.h.

Referenced by ng_bind(), ng_bridge_nodename(), ng_btsocket_hci_raw_control(), ng_h4_open(), ng_h4_shutdown(), and ng_name_node().

7.122.1.25 #define NG_PATHLEN (NG_PATHSIZ - 1)

Definition at line 59 of file ng_message.h.

7.122.1.26 #define NG_PATHSIZ 512

Definition at line 51 of file ng_message.h.

Referenced by ng_path2noderef().

7.122.1.27 #define NG_TEXTRESPONSE 1024

Definition at line 63 of file ng_message.h.

Referenced by cisco_rcvmsg(), ng_atm_rcvmsg(), ng_bt3c_rcvmsg(), ng_h4_rcvmsg(), ng_hci_default_rcvmsg(), ng_l2cap_default_rcvmsg(), ng_sscfu_rcvmsg(), ng_sscop_rcvmsg(), ng_ibt_rcvmsg(), ng_uni_rcvmsg(), and nglmi_rcvmsg().

7.122.1.28 #define NG_TYPELEN (NG_TYPESIZ - 1)

Definition at line 56 of file ng_message.h.

7.122.1.29 #define NG_TYPESIZ 32

Definition at line 48 of file ng_message.h.

Referenced by ng_newtype(), and ngc_send().

7.122.1.30 #define NG_VERSION 8

Definition at line 106 of file ng_message.h.

Referenced by ngc_send().

7.122.1.31 #define NGF_ORIG 0x00000000

Definition at line 109 of file ng_message.h.

7.122.1.32 #define NGF_RESP 0x00000001

Definition at line 110 of file ng_message.h.

Referenced by `cisco_rcvmsg()`, `ng_apply_item()`, `ng_atmllc_rcvmsg()`, `ng_btsocket_hci_raw_node_rcvmsg()`, `ng_btsocket_l2cap_raw_node_rcvmsg()`, `ng_generic_msg()`, `ng_ksocket_finish_accept()`, `ng_ksocket_incoming2()`, `ng_l2cap_l2ca_cfg_rsp()`, `ng_l2cap_l2ca_cfg_rsp_rsp()`, `ng_l2cap_l2ca_con_rsp()`, `ng_l2cap_l2ca_con_rsp_rsp()`, `ng_l2cap_l2ca_discon_rsp()`, `ng_l2cap_l2ca_enable_clt()`, `ng_l2cap_l2ca_get_info_rsp()`, `ng_l2cap_l2ca_ping_rsp()`, `ng_l2cap_l2ca_write_rsp()`, `ng_pppoe_rcvmsg()`, `ng_source_rcvmsg()`, and `nge_rcvmsg()`.

7.122.1.33 #define NGIOCGINFO_IOR('N', 40, struct nodeinfo)

Definition at line 380 of file ng_message.h.

Referenced by `ng_h4_ioctl()`, and `ngt_tioctl()`.

7.122.1.34 #define NGIOCSETNAME_IOW('N', 41, struct ngm_name)

Definition at line 381 of file ng_message.h.

7.122.1.35 #define NGM_DROP_LINK 41

Definition at line 171 of file ng_message.h.

7.122.1.36 #define NGM_FLOW_COOKIE 851672669

Definition at line 161 of file ng_message.h.

Referenced by `cisco_notify()`, `ng_atm_event_func()`, `ng_eiface_rcvmsg()`, `ng_ether_link_state()`, `ng_iface_rcvmsg()`, `ng_one2many_notify()`, `ng_one2many_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_UI_rcvmsg()`, `ng_vlan_rcvmsg()`, and `ngt_rcvmsg()`.

7.122.1.37 #define NGM_FLUSH_QUEUE 43

Definition at line 173 of file ng_message.h.

7.122.1.38 #define NGM_GENERIC_COOKIE 1137070366

Definition at line 122 of file ng_message.h.

Referenced by `cisco_rcvmsg()`, `ng_apply_item()`, `ng_atm_rcvmsg()`, `ng_bt3c_rcvmsg()`, `ng_btsocket_hci_raw_control()`, `ng_btsocket_hci_raw_node_rcvmsg()`, `ng_generic_msg()`, `ng_h4_rcvmsg()`, `ng_hci_default_rcvmsg()`, `ng_l2cap_default_rcvmsg()`, `ng_sscfu_rcvmsg()`, `ng_sscop_rcvmsg()`, `ng_ubt_rcvmsg()`, `ng_uni_rcvmsg()`, `ngc_send()`, `nglmi_inquire()`, and `nglmi_rcvmsg()`.

7.122.1.39 #define NGM_GET_BANDWIDTH (44|NGM_READONLY)

Definition at line 174 of file ng_message.h.

7.122.1.40 #define NGM_GET_XMIT_Q_LIMITS (46|NGM_READONLY)

Definition at line 176 of file ng_message.h.

7.122.1.41 #define NGM_HASREPLY 0x20000000

Definition at line 84 of file ng_message.h.

7.122.1.42 #define NGM_HIGH_WATER_PASSED 34

Definition at line 166 of file ng_message.h.

Referenced by flow_lower(), flow_upper(), and ng_atm_event_func().

7.122.1.43 #define NGM_LINK_IS_DOWN 33

Definition at line 165 of file ng_message.h.

Referenced by cisco_keepalive(), ng_eiface_rcvmsg(), ng_ether_link_state(), ng_iface_rcvmsg(), and ng_one2many_rcvmsg().

7.122.1.44 #define NGM_LINK_IS_UP 32

Definition at line 164 of file ng_message.h.

Referenced by cisco_input(), ng_eiface_rcvmsg(), ng_ether_link_state(), ng_iface_rcvmsg(), and ng_one2many_rcvmsg().

7.122.1.45 #define NGM_LOW_WATER_PASSED 35

Definition at line 167 of file ng_message.h.

Referenced by flow_lower(), flow_upper(), and ng_atm_event_func().

7.122.1.46 #define NGM_MICROMANAGE 47

Definition at line 177 of file ng_message.h.

7.122.1.47 #define NGM_RAISE_LINK 42

Definition at line 172 of file ng_message.h.

7.122.1.48 #define NGM_READONLY 0x10000000

Definition at line 82 of file ng_message.h.

Referenced by ng_package_msg().

7.122.1.49 #define NGM_SET_FLOW_MANAGER 48

Definition at line 179 of file ng_message.h.

7.122.1.50 #define NGM_SET_XMIT_Q_LIMITS 45

Definition at line 175 of file ng_message.h.

7.122.1.51 #define NGM_SYNC_QUEUE_STATE 36

Definition at line 168 of file ng_message.h.

Referenced by flow_upper().

7.122.2 Enumeration Type Documentation**7.122.2.1 anonymous enum**

Enumerator:

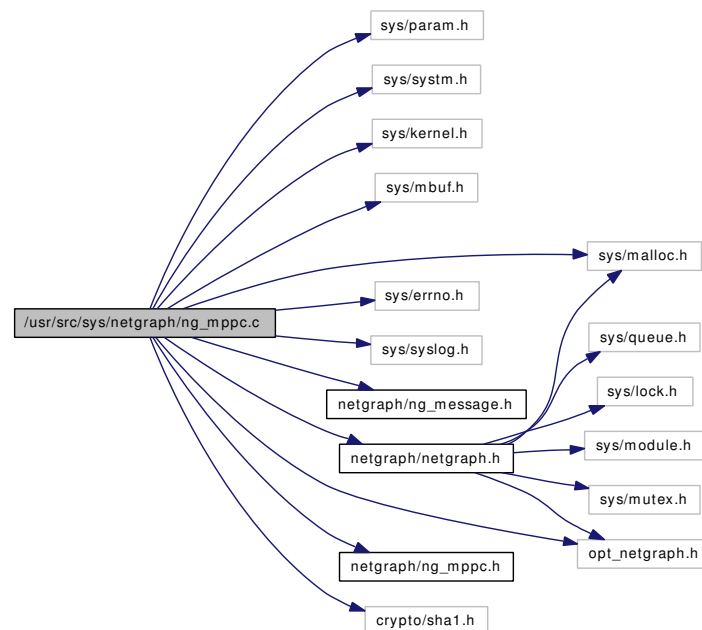
NGM_SHUTDOWN
NGM_MKPEER
NGM_CONNECT
NGM_NAME
NGM_RMHOOK
NGM_NODEINFO
NGM_LISTHOOKS
NGM_LISTNAMES
NGM_LISTNODES
NGM_LISTTYPES
NGM_TEXT_STATUS
NGM_BINARY2ASCII
NGM_ASCII2BINARY
NGM_TEXT_CONFIG

Definition at line 125 of file ng_message.h.

7.123 /usr/src/sys/netgraph/ng_mppc.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_mppc.h>
#include "opt_netgraph.h"
#include <crypto/sha1.h>
```

Include dependency graph for ng_mppc.c:



Data Structures

- struct [ng_mppc_dir](#)
- struct [ng_mppc_private](#)

Defines

- #define [M_NETGRAPH_MPPC](#) M_NETGRAPH

- #define `MPPC_DECOMP_BUFSIZE` 8092
- #define `MPPC_DECOMP_SAFETY` 100
- #define `MPPC_HDRLEN` 2
- #define `KEYLEN(b)` (((b) & MPPE_128) ? 16 : 8)
- #define `MPPE_MAX_REKEY` 1000
- #define `MPPC_FLAG_FLUSHED` 0x8000
- #define `MPPC_FLAG_RESTART` 0x4000
- #define `MPPC_FLAG_COMPRESSED` 0x2000
- #define `MPPC_FLAG_ENCRYPTED` 0x1000
- #define `MPPC_CCOUNT_MASK` 0x0fff
- #define `MPPE_UPDATE_MASK` 0xff
- #define `MPPE_UPDATE_FLAG` 0xff
- #define `MPPC_COMP_OK` 0x05
- #define `MPPC_DECOMP_OK` 0x05
- #define `ERRROUT(x)` do { error = (x); goto done; } while (0)

Typedefs

- typedef `ng_mppc_private` * `priv_p`

Functions

- static int `ng_mppc_compress` (`node_p` node, struct mbuf *m, struct mbuf **resultp)
- static int `ng_mppc_decompress` (`node_p` node, struct mbuf *m, struct mbuf **resultp)
- static void `ng_mppc_getkey` (const u_char *h, u_char *h2, int len)
- static void `ng_mppc_updatekey` (u_int32_t bits, u_char *key0, u_char *key, struct rc4_state *rc4)
- static void `ng_mppc_reset_req` (`node_p` node)
- `NETGRAPH_INIT` (mppc,&ng_mppc_typestruct)
- static int `ng_mppc_constructor` (`node_p` node)
- static int `ng_mppc_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_mppc_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_mppc_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_mppc_shutdown` (`node_p` node)
- static int `ng_mppc_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_mppc_constructor`
- static `ng_rcvmsg_t` `ng_mppc_rcvmsg`
- static `ng_shutdown_t` `ng_mppc_shutdown`
- static `ng_newhook_t` `ng_mppc_newhook`
- static `ng_rcvdata_t` `ng_mppc_rcvdata`
- static `ng_disconnect_t` `ng_mppc_disconnect`
- static struct `ng_type` `ng_mppc_typestruct`
- static const u_char `ng_mppe_weakenkey` [3] = { 0xd1, 0x26, 0x9e }

7.123.1 Define Documentation

7.123.1.1 **#define ERROUT(x) do { error = (x); goto done; } while (0)**

Definition at line 184 of file ng_mppc.c.

7.123.1.2 **#define KEYLEN(b) (((b) & MPPE_128) ? 16 : 8)**

Definition at line 98 of file ng_mppc.c.

Referenced by ng_mppc_compress(), ng_mppc_decompress(), ng_mppc_rcvmsg(), ng_mppc_reset_req(), and ng_mppc_updatekey().

7.123.1.3 **#define M_NETGRAPH_MPPC M_NETGRAPH**

Definition at line 78 of file ng_mppc.c.

Referenced by ng_mppc_compress(), ng_mppc_constructor(), ng_mppc_decompress(), ng_mppc_rcvmsg(), and ng_mppc_shutdown().

7.123.1.4 **#define MPPC_CCOUNT_MASK 0x0fff**

Definition at line 114 of file ng_mppc.c.

Referenced by ng_mppc_decompress().

7.123.1.5 **#define MPPC_COMP_OK 0x05**

Definition at line 119 of file ng_mppc.c.

Referenced by ng_mppc_compress().

7.123.1.6 **#define MPPC_DECOMP_BUFSIZE 8092**

Definition at line 91 of file ng_mppc.c.

Referenced by ng_mppc_decompress().

7.123.1.7 **#define MPPC_DECOMP_OK 0x05**

Definition at line 120 of file ng_mppc.c.

Referenced by ng_mppc_decompress().

7.123.1.8 **#define MPPC_DECOMP_SAFETY 100**

Definition at line 92 of file ng_mppc.c.

Referenced by ng_mppc_decompress().

7.123.1.9 #define MPPC_FLAG_COMPRESSED 0x2000

Definition at line 112 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.10 #define MPPC_FLAG_ENCRYPTED 0x1000

Definition at line 113 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.11 #define MPPC_FLAG_FLUSHED 0x8000

Definition at line 110 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.12 #define MPPC_FLAG_RESTART 0x4000

Definition at line 111 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.13 #define MPPC_HDRLEN 2

Definition at line 95 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.14 #define MPPE_MAX_REKEY 1000

Definition at line 107 of file ng_mppc.c.

Referenced by ng_mppc_decompress().

7.123.1.15 #define MPPE_UPDATE_FLAG 0xff

Definition at line 117 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.1.16 #define MPPE_UPDATE_MASK 0xff

Definition at line 116 of file ng_mppc.c.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

7.123.2 Typedef Documentation

7.123.2.1 typedef struct [ng_mppc_private*](#) [priv_p](#)

Definition at line 143 of file `ng_mppc.c`.

7.123.3 Function Documentation

7.123.3.1 NETGRAPH_INIT (mppc, & [ng_mppc_typestruct](#))

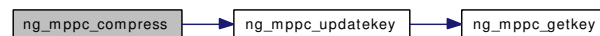
7.123.3.2 static int [ng_mppc_compress](#) ([node_p node](#), struct mbuf * *m*, struct mbuf ** *resultp*) [static]

Definition at line 467 of file `ng_mppc.c`.

References `ng_mppc_config::bits`, `ng_mppc_dir::cc`, `ng_mppc_dir::cfg`, `ng_mppc_dir::flushed`, `KEYLEN`, `M_NETGRAPH_MPPC`, `MPPC_BIT`, `MPPC_COMP_OK`, `MPPC_FLAG_COMPRESSED`, `MPPC_FLAG_ENCRYPTED`, `MPPC_FLAG_FLUSHED`, `MPPC_FLAG_RESTART`, `MPPC_HDRLEN`, `MPPC_MAX_BLOWUP`, `MPPE_BITS`, `MPPE_STATELESS`, `MPPE_UPDATE_FLAG`, `MPPE_UPDATE_MASK`, `ng_mppc_updatekey()`, `NG_NODE_PRIVATE`, and `ng_mppc_config::startkey`.

Referenced by `ng_mppc_rcvdata()`.

Here is the call graph for this function:



7.123.3.3 static int [ng_mppc_constructor](#) ([node_p node](#)) [static]

Definition at line 194 of file `ng_mppc.c`.

References `M_NETGRAPH_MPPC`, `NG_NODE_FORCE_WRITER`, and `NG_NODE_SET_PRIVATE`.

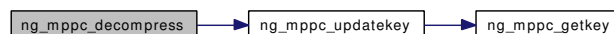
7.123.3.4 static int [ng_mppc_decompress](#) ([node_p node](#), struct mbuf * *m*, struct mbuf ** *resultp*) [static]

Definition at line 583 of file `ng_mppc.c`.

References `ng_mppc_config::bits`, `ng_mppc_dir::cc`, `ng_mppc_dir::cfg`, `KEYLEN`, `M_NETGRAPH_MPPC`, `MPPC_BIT`, `MPPC_CCOUNT_MASK`, `MPPC_DECOMP_BUFSIZE`, `MPPC_DECOMP_OK`, `MPPC_DECOMP_SAFETY`, `MPPC_FLAG_COMPRESSED`, `MPPC_FLAG_ENCRYPTED`, `MPPC_FLAG_FLUSHED`, `MPPC_FLAG_RESTART`, `MPPC_HDRLEN`, `MPPE_BITS`, `MPPE_MAX_REKEY`, `MPPE_STATELESS`, `MPPE_UPDATE_FLAG`, `MPPE_UPDATE_MASK`, `ng_mppc_updatekey()`, `NG_NODE_PRIVATE`, and `ng_mppc_config::startkey`.

Referenced by `ng_mppc_rcvdata()`.

Here is the call graph for this function:

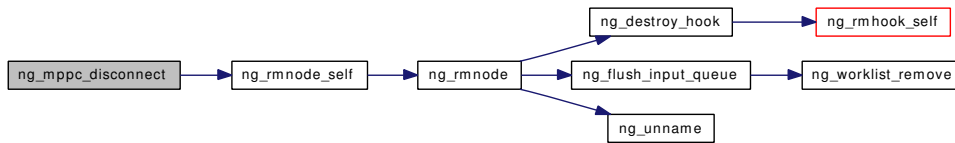


7.123.3.5 static int ng_mppc_disconnect (**hook_p** hook) [static]

Definition at line 440 of file ng_mppc.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.123.3.6 static void ng_mppc_getkey (const u_char * h, u_char * h2, int len) [static]

Definition at line 774 of file ng_mppc.c.

Referenced by ng_mppc_rcvmsg(), and ng_mppc_updatekey().

7.123.3.7 static int ng_mppc_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 216 of file ng_mppc.c.

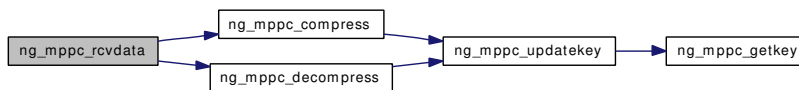
References NG_MPPC_HOOK_COMP, NG_MPPC_HOOK_DECOMP, and NG_NODE_PRIVATE.

7.123.3.8 static int ng_mppc_rcvdata (hook_p hook, item_p item) [static]

Definition at line 353 of file ng_mppc.c.

References NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_MKMESSAGE, ng_mppc_compress(), ng_mppc_decompress(), NG_NODE_PRIVATE, NG_SEND_MSG_ID, NGI_GET_M, NGM_MPPC_COOKIE, and NGM_MPPC_RESETREQ.

Here is the call graph for this function:

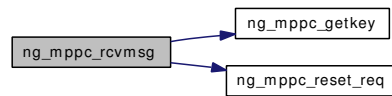


7.123.3.9 static int ng_mppc_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 242 of file ng_mppc.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERRROUT, ng_mesg::header, KEYLEN, M_NETGRAPH_MPPC, MPPC_BIT, MPPC_VALID_BITS, MPPE_40, MPPE_56, MPPE_BITS, NG_FREE_MSG, ng_mppc_getkey(), ng_mppc_reset_req(), ng_mppe_weakenkey, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGI_RETADDR, NGM_MPPC_CONFIG_COMP, NGM_MPPC_CONFIG_DECOMP, NGM_MPPC_COOKIE, NGM_MPPC_RESETREQ, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



7.123.3.10 static void ng_mppc_reset_req (node_p node) [static]

Definition at line 754 of file ng_mppc.c.

References ng_mppc_config::bits, ng_mppc_dir::cfg, ng_mppc_dir::flushed, KEYLEN, MPPE_STATELESS, and NG_NODE_PRIVATE.

Referenced by ng_mppc_rcvmsg().

7.123.3.11 static int ng_mppc_shutdown (node_p node) [static]

Definition at line 418 of file ng_mppc.c.

References M_NETGRAPH_MPPC, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.123.3.12 static void ng_mppc_updatekey (u_int32_t bits, u_char * key0, u_char * key, struct rc4_state * rc4) [static]

Definition at line 800 of file ng_mppc.c.

References KEYLEN, MPPE_40, MPPE_56, ng_mppc_getkey(), and ng_mppe_weakenkey.

Referenced by ng_mppc_compress(), and ng_mppc_decompress().

Here is the call graph for this function:



7.123.4 Variable Documentation

7.123.4.1 ng_constructor_t ng_mppc_constructor [static]

Definition at line 146 of file ng_mppc.c.

7.123.4.2 ng_disconnect_t ng_mppc_disconnect [static]

Definition at line 151 of file ng_mppc.c.

7.123.4.3 ng_newhook_t ng_mppc_newhook [static]

Definition at line 149 of file ng_mppc.c.

7.123.4.4 `ng_rcvdata_t ng_mppc_rcvdata` [static]

Definition at line 150 of file `ng_mppc.c`.

7.123.4.5 `ng_rcvmsg_t ng_mppc_rcvmsg` [static]

Definition at line 147 of file `ng_mppc.c`.

7.123.4.6 `ng_shutdown_t ng_mppc_shutdown` [static]

Definition at line 148 of file `ng_mppc.c`.

7.123.4.7 `struct ng_type ng_mppc_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_MPPC_NODE_TYPE,
    .constructor = ng_mppc_constructor,
    .rcvmsg =      ng_mppc_rcvmsg,
    .shutdown =    ng_mppc_shutdown,
    .newhook =     ng_mppc_newhook,
    .rcvdata =     ng_mppc_rcvdata,
    .disconnect =  ng_mppc_disconnect,
}
```

Definition at line 164 of file `ng_mppc.c`.

7.123.4.8 `const u_char ng_mppe_weakenkey[3] = { 0xd1, 0x26, 0x9e }` [static]

Definition at line 182 of file `ng_mppc.c`.

Referenced by `ng_mppc_rcvmsg()`, and `ng_mppc_updatekey()`.

7.124 /usr/src/sys/netgraph/ng_mppc.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_mppc_config](#)

Defines

- #define [NG_MPPC_NODE_TYPE](#) "mppc"
- #define [NGM_MPPC_COOKIE](#) 942886745
- #define [NG_MPPC_HOOK_COMP](#) "comp"
- #define [NG_MPPC_HOOK_DECOMP](#) "decomp"
- #define [MPPE_KEY_LEN](#) 16
- #define [MPPC_MAX_BLOWUP\(n\)](#) ((n) * 9 / 8 + 26)
- #define [MPPC_BIT](#) 0x00000001
- #define [MPPE_40](#) 0x00000020
- #define [MPPE_56](#) 0x00000080
- #define [MPPE_128](#) 0x00000040
- #define [MPPE_BITS](#) 0x000000e0
- #define [MPPE_STATELESS](#) 0x01000000
- #define [MPPC_VALID_BITS](#) 0x010000e1

Enumerations

- enum { [NGM_MPPC_CONFIG_COMP](#) = 1, [NGM_MPPC_CONFIG_DECOMP](#), [NGM_MPPC_RESETREQ](#) }

7.124.1 Define Documentation

7.124.1.1 #define MPPC_BIT 0x00000001

Definition at line 62 of file [ng_mppc.h](#).

Referenced by [ng_mppc_compress\(\)](#), [ng_mppc_decompress\(\)](#), and [ng_mppc_rcvmsg\(\)](#).

7.124.1.2 #define MPPC_MAX_BLOWUP(n) ((n) * 9 / 8 + 26)

Definition at line 59 of file [ng_mppc.h](#).

Referenced by [ng_mppc_compress\(\)](#).

7.124.1.3 #define MPPC_VALID_BITS 0x010000e1

Definition at line 68 of file ng_mppc.h.

Referenced by ng_mppc_rcvmsg().

7.124.1.4 #define MPPE_128 0x00000040

Definition at line 65 of file ng_mppc.h.

7.124.1.5 #define MPPE_40 0x00000020

Definition at line 63 of file ng_mppc.h.

Referenced by ng_mppc_rcvmsg(), and ng_mppc_updatekey().

7.124.1.6 #define MPPE_56 0x00000080

Definition at line 64 of file ng_mppc.h.

Referenced by ng_mppc_rcvmsg(), and ng_mppc_updatekey().

7.124.1.7 #define MPPE_BITS 0x000000e0

Definition at line 66 of file ng_mppc.h.

Referenced by ng_mppc_compress(), ng_mppc_decompress(), and ng_mppc_rcvmsg().

7.124.1.8 #define MPPE_KEY_LEN 16

Definition at line 56 of file ng_mppc.h.

7.124.1.9 #define MPPE_STATELESS 0x01000000

Definition at line 67 of file ng_mppc.h.

Referenced by ng_mppc_compress(), ng_mppc_decompress(), and ng_mppc_reset_req().

7.124.1.10 #define NG_MPPC_HOOK_COMP "comp"

Definition at line 52 of file ng_mppc.h.

Referenced by ng_mppc_newhook().

7.124.1.11 #define NG_MPPC_HOOK_DECOMP "decomp"

Definition at line 53 of file ng_mppc.h.

Referenced by ng_mppc_newhook().

7.124.1.12 #define NG_MPPC_NODE_TYPE "mppc"

Definition at line 48 of file ng_mppc.h.

7.124.1.13 #define NGM_MPPC_COOKIE 942886745

Definition at line 49 of file ng_mppc.h.

Referenced by ng_mppc_rcvdata(), and ng_mppc_rcvmsg().

7.124.2 Enumeration Type Documentation**7.124.2.1 anonymous enum**

Enumerator:

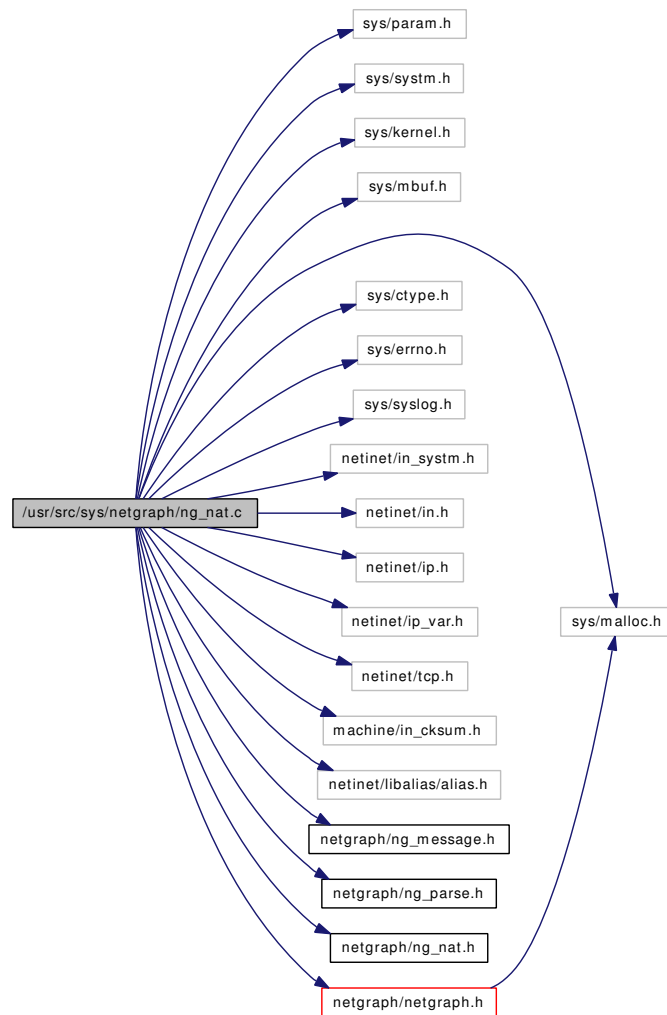
NGM_MPPC_CONFIG_COMP
NGM_MPPC_CONFIG_DECOMP
NGM_MPPC_RESETRQ

Definition at line 78 of file ng_mppc.h.

7.125 /usr/src/sys/netgraph/ng_nat.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <netinet/in_system.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/tcp.h>
#include <machine/in_cksum.h>
#include <netinet/libalias/alias.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_nat.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_nat.c:



Data Structures

- struct [ng_nat_priv](#)

Defines

- #define [NGNAT_READY](#) 0x1
- #define [NGNAT_ADDR_DEFINED](#) 0x2

Typedefs

- typedef [ng_nat_priv](#) * [priv_p](#)

Functions

- [NETGRAPH_INIT](#) (nat,&typestruct)
- [MODULE_DEPEND](#) (ng_nat, libalias, 1, 1, 1)

- static int `ng_nat_constructor` (`node_p` node)
- static int `ng_nat_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_nat_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_nat_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_nat_shutdown` (`node_p` node)
- static int `ng_nat_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_nat_constructor`
- static `ng_rcvmsg_t` `ng_nat_rcvmsg`
- static `ng_shutdown_t` `ng_nat_shutdown`
- static `ng_newhook_t` `ng_nat_newhook`
- static `ng_rcvdata_t` `ng_nat_rcvdata`
- static `ng_disconnect_t` `ng_nat_disconnect`
- static struct `ng_cmdlist` `ng_nat_cmdlist` []
- static struct `ng_type` `typestruct`

7.125.1 Define Documentation

7.125.1.1 #define NGNAT_ADDR_DEFINED 0x2

Definition at line 98 of file `ng_nat.c`.

Referenced by `ng_nat_newhook()`, and `ng_nat_rcvmsg()`.

7.125.1.2 #define NGNAT_READY 0x1

Definition at line 97 of file `ng_nat.c`.

Referenced by `ng_nat_disconnect()`, `ng_nat_newhook()`, `ng_nat_rcvdata()`, and `ng_nat_rcvmsg()`.

7.125.2 Typedef Documentation

7.125.2.1 typedef struct `ng_nat_priv*` `priv_p`

Definition at line 94 of file `ng_nat.c`.

7.125.3 Function Documentation

7.125.3.1 MODULE_DEPEND (`ng_nat`, `libalias`, 1, 1, 1)

7.125.3.2 NETGRAPH_INIT (`nat`, & `typestruct`)

7.125.3.3 static int `ng_nat_constructor` (`node_p` `node`) [static]

Definition at line 101 of file `ng_nat.c`.

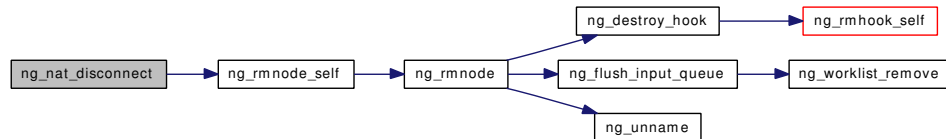
References `NG_NODE_FORCE_WRITER`, `NG_NODE_SET_PRIVATE`, and `ng_nat_priv::node`.

7.125.3.4 static int ng_nat_disconnect (hook_p hook) [static]

Definition at line 313 of file ng_nat.c.

References NG_HOOK_NODE, NG_NODE_PRIVATE, ng_rmnode_self(), and NGNAT_READY.

Here is the call graph for this function:

**7.125.3.5** static int ng_nat_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 136 of file ng_nat.c.

References NG_NAT_HOOK_IN, NG_NAT_HOOK_OUT, NG_NODE_PRIVATE, NGNAT_ADDR_DEFINED, NGNAT_READY, and ng_nat_priv::node.

7.125.3.6 static int ng_nat_rcvdata (hook_p hook, item_p item) [static]

Definition at line 201 of file ng_nat.c.

References NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_HOOK_NODE, NG_NODE_PRIVATE, NGL_M, and NGNAT_READY.

7.125.3.7 static int ng_nat_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 156 of file ng_nat.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ng_mesg::header, NG_FREE_MSG, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGL_GET_MSG, NGM_NAT_COOKIE, NGM_NAT_SET_IPADDR, NGNAT_ADDR_DEFINED, NGNAT_READY, ng_nat_priv::node, and ng_mesg::ng_msghdr::typecookie.

7.125.3.8 static int ng_nat_shutdown (node_p node) [static]

Definition at line 300 of file ng_nat.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.125.4 Variable Documentation**7.125.4.1** struct ng_cmdlist ng_nat_cmdlist[] [static]

Initial value:

```

{
    {
        NGM_NAT_COOKIE,
    }
}
  
```

```

    NGM_NAT_SET_IPADDR,
    "setaliasaddr",
    &ng_parse_ipaddr_type,
    NULL
},
{ 0 }
}

```

Definition at line 60 of file ng_nat.c.

7.125.4.2 [ng_constructor_t ng_nat_constructor](#) [static]

Definition at line 52 of file ng_nat.c.

7.125.4.3 [ng_disconnect_t ng_nat_disconnect](#) [static]

Definition at line 57 of file ng_nat.c.

7.125.4.4 [ng_newhook_t ng_nat_newhook](#) [static]

Definition at line 55 of file ng_nat.c.

7.125.4.5 [ng_rcvdata_t ng_nat_rcvdata](#) [static]

Definition at line 56 of file ng_nat.c.

7.125.4.6 [ng_rcvmsg_t ng_nat_rcvmsg](#) [static]

Definition at line 53 of file ng_nat.c.

7.125.4.7 [ng_shutdown_t ng_nat_shutdown](#) [static]

Definition at line 54 of file ng_nat.c.

7.125.4.8 [struct ng_type typestruct](#) [static]

Initial value:

```

{
    .version =      NG_ABI_VERSION,
    .name =        NG_NAT_NODE_TYPE,
    .constructor = ng_nat_constructor,
    .rcvmsg =      ng_nat_rcvmsg,
    .shutdown =    ng_nat_shutdown,
    .newhook =     ng_nat_newhook,
    .rcvdata =     ng_nat_rcvdata,
    .disconnect =  ng_nat_disconnect,
    .cmdlist =     ng_nat_cmdlist,
}

```

Definition at line 72 of file ng_nat.c.

7.126 /usr/src/sys/netgraph/ng_nat.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_NAT_NODE_TYPE](#) "nat"
- #define [NGM_NAT_COOKIE](#) 1107718711
- #define [NG_NAT_HOOK_IN](#) "in"
- #define [NG_NAT_HOOK_OUT](#) "out"

Enumerations

- enum { [NGM_NAT_SET_IPADDR](#) = 1 }

7.126.1 Define Documentation

7.126.1.1 #define [NG_NAT_HOOK_IN](#) "in"

Definition at line 32 of file ng_nat.h.

Referenced by ng_nat_newhook().

7.126.1.2 #define [NG_NAT_HOOK_OUT](#) "out"

Definition at line 33 of file ng_nat.h.

Referenced by ng_nat_newhook().

7.126.1.3 #define [NG_NAT_NODE_TYPE](#) "nat"

Definition at line 29 of file ng_nat.h.

7.126.1.4 #define [NGM_NAT_COOKIE](#) 1107718711

Definition at line 30 of file ng_nat.h.

Referenced by ng_nat_rcvmsg().

7.126.2 Enumeration Type Documentation

7.126.2.1 anonymous enum

Enumerator:

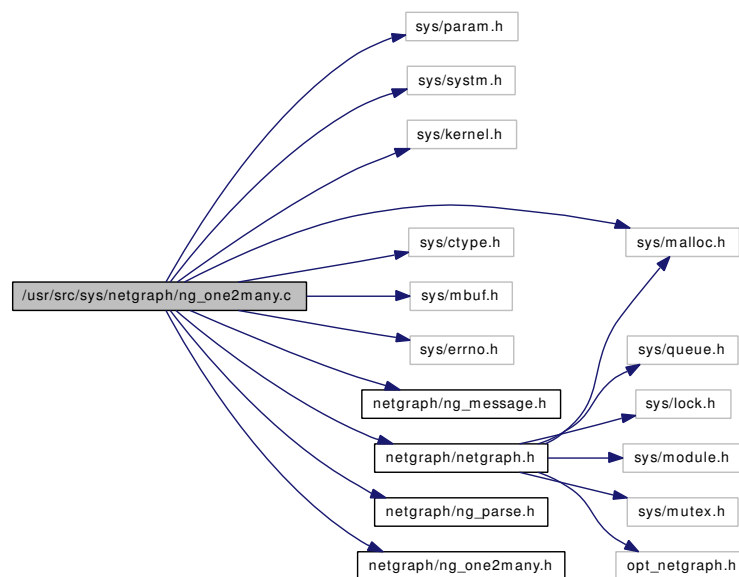
NGM_NAT_SET_IPADDR

Definition at line 35 of file ng_nat.h.

7.127 /usr/src/sys/netgraph/ng_one2many.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_one2many.h>
```

Include dependency graph for ng_one2many.c:



Data Structures

- struct [ng_one2many_link](#)
- struct [ng_one2many_private](#)

Typedefs

- typedef [ng_one2many_private](#) * `priv_p`

Functions

- static void `ng_one2many_update_many` (`priv_p` `priv`)
- static void `ng_one2many_notify` (`priv_p` `priv`, `uint32_t` `cmd`)
- `NETGRAPH_INIT` (`one2many`, &`ng_one2many_tpestruct`)
- static int `ng_one2many_constructor` (`node_p` `node`)
- static int `ng_one2many_newhook` (`node_p` `node`, `hook_p` `hook`, `const char *name`)
- static int `ng_one2many_rcvmsg` (`node_p` `node`, `item_p` `item`, `hook_p` `lasthook`)
- static int `ng_one2many_rcvdata` (`hook_p` `hook`, `item_p` `item`)
- static int `ng_one2many_shutdown` (`node_p` `node`)
- static int `ng_one2many_disconnect` (`hook_p` `hook`)

Variables

- static `ng_constructor_t` `ng_one2many_constructor`
- static `ng_rcvmsg_t` `ng_one2many_rcvmsg`
- static `ng_shutdown_t` `ng_one2many_shutdown`
- static `ng_newhook_t` `ng_one2many_newhook`
- static `ng_rcvdata_t` `ng_one2many_rcvdata`
- static `ng_disconnect_t` `ng_one2many_disconnect`
- static struct `ng_parse_fixedarray_info` `ng_one2many_enableLinks_array_type_info`
- static struct `ng_parse_type` `ng_one2many_enableLinks_array_type`
- static struct `ng_parse_struct_field` `ng_one2many_config_type_fields` [] = `NG_ONE2MANY_CONFIG_TYPE_INFO`(&`ng_one2many_enableLinks_array_type`)
- static struct `ng_parse_type` `ng_one2many_config_type`
- static struct `ng_parse_struct_field` `ng_one2many_link_stats_type_fields` [] = `NG_ONE2MANY_LINK_STATS_TYPE_INFO`
- static struct `ng_parse_type` `ng_one2many_link_stats_type`
- static struct `ng_cmdlist` `ng_one2many_cmdlist` []
- static struct `ng_type` `ng_one2many_tpestruct`

7.127.1 Typedef Documentation

7.127.1.1 typedef struct `ng_one2many_private*` `priv_p`

Definition at line 80 of file `ng_one2many.c`.

7.127.2 Function Documentation

7.127.2.1 `NETGRAPH_INIT` (`one2many`, & `ng_one2many_tpestruct`)

7.127.2.2 static int `ng_one2many_constructor` (`node_p` `node`) [static]

Definition at line 185 of file `ng_one2many.c`.

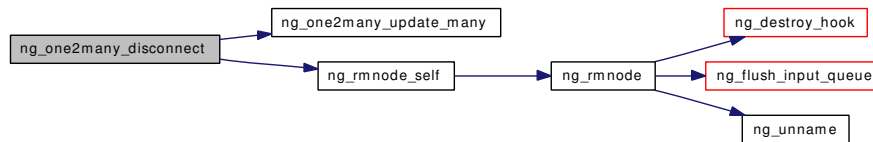
References `NG_NODE_SET_PRIVATE`, `NG_ONE2MANY_FAIL_MANUAL`, and `NG_ONE2MANY_XMIT_ROUNDROBIN`.

7.127.2.3 static int ng_one2many_disconnect (hook_p hook) [static]

Definition at line 514 of file ng_one2many.c.

References `ng_one2many_link::hook`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_IS_INVALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, `NG_ONE2MANY_MAX_LINKS`, `NG_ONE2MANY_ONE_LINKNUM`, `ng_one2many_update_many()`, and `ng_rmnode_self()`.

Here is the call graph for this function:

**7.127.2.4 static int ng_one2many_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 208 of file ng_one2many.c.

References `ng_one2many_link::hook`, `NG_HOOK_SET_PRIVATE`, `NG_NODE_PRIVATE`, `NG_ONE2MANY_HOOK_MANY_PREFIX`, `NG_ONE2MANY_HOOK_ONE`, `NG_ONE2MANY_MAX_LINKS`, `NG_ONE2MANY_ONE_LINKNUM`, `ng_one2many_update_many()`, and `ng_one2many_link::stats`.

Here is the call graph for this function:

**7.127.2.5 static void ng_one2many_notify (priv_p priv, uint32_t cmd) [static]**

Definition at line 598 of file ng_one2many.c.

References `NG_MKMESSAGE`, `NG_SEND_MSG_HOOK`, and `NGM_FLOW_COOKIE`.

7.127.2.6 static int ng_one2many_rcvdata (hook_p hook, item_p item) [static]

Definition at line 413 of file ng_one2many.c.

References `ng_one2many_link::hook`, `ng_one2many_link_stats::memoryFailures`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_ITEM_HOOK`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `NG_ONE2MANY_MAX_LINKS`, `NG_ONE2MANY_ONE_LINKNUM`, `NG_ONE2MANY_XMIT_ALL`, `NG_ONE2MANY_XMIT_ROUNDROBIN`, `NG_SEND_DATA_ONLY`, `NGI_M`, `ng_one2many_link_stats::recvOctets`, `ng_one2many_link_stats::recvPackets`, `ng_one2many_link::stats`, `ng_one2many_link_stats::xmitOctets`, and `ng_one2many_link_stats::xmitPackets`.

7.127.2.7 static int ng_one2many_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 256 of file ng_one2many.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_HOOK_PRIVATE`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_ONE2MANY_FAIL_MANUAL`, `NG_ONE2MANY_FAIL_NOTIFY`, `NG_ONE2MANY_MAX_LINKS`, `NG_ONE2MANY_ONE_LINKNUM`, `ng_one2many_update_many()`, `NG_ONE2MANY_XMIT_ALL`, `NG_ONE2MANY_XMIT_ROUNDROBIN`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_FLOW_COOKIE`, `NGM_LINK_IS_DOWN`, `NGM_LINK_IS_UP`, `NGM_ONE2MANY_CLR_STATS`, `NGM_ONE2MANY_COOKIE`, `NGM_ONE2MANY_GET_CONFIG`, `NGM_ONE2MANY_GET_STATS`, `NGM_ONE2MANY_GETCLR_STATS`, `NGM_ONE2MANY_SET_CONFIG`, `ng_one2many_link::stats`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.127.2.8 static int `ng_one2many_shutdown` (`node_p node`) [static]

Definition at line 498 of file `ng_one2many.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.127.2.9 static void `ng_one2many_update_many` (`priv_p priv`) [static]

Definition at line 549 of file `ng_one2many.c`.

References `NG_ONE2MANY_FAIL_MANUAL`, `NG_ONE2MANY_FAIL_NOTIFY`, and `NG_ONE2MANY_MAX_LINKS`.

Referenced by `ng_one2many_disconnect()`, `ng_one2many_newhook()`, and `ng_one2many_rcvmsg()`.

7.127.3 Variable Documentation

7.127.3.1 struct `ng_cmdlist ng_one2many_cmdlist`[] [static]

Definition at line 124 of file `ng_one2many.c`.

7.127.3.2 struct `ng_parse_type ng_one2many_config_type` [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_one2many_config_type_fields
}
  
```

Definition at line 110 of file `ng_one2many.c`.

7.127.3.3 struct `ng_parse_struct_field ng_one2many_config_type_fields`[] = `NG_ONE2MANY_CONFIG_TYPE_INFO(&ng_one2many_enableLinks_array_type)` [static]

Definition at line 109 of file `ng_one2many.c`.

7.127.3.4 `ng_constructor_t ng_one2many_constructor` [static]

Definition at line 83 of file ng_one2many.c.

7.127.3.5 `ng_disconnect_t ng_one2many_disconnect` [static]

Definition at line 88 of file ng_one2many.c.

7.127.3.6 `struct ng_parse_type ng_one2many_enableLinks_array_type` [static]**Initial value:**

```
{
    &ng_parse_fixedarray_type,
    &ng_one2many_enableLinks_array_type_info,
}
```

Definition at line 104 of file ng_one2many.c.

7.127.3.7 `struct ng_parse_fixedarray_info ng_one2many_enableLinks_array_type_info`
[static]**Initial value:**

```
{
    &ng_parse_uint8_type,
    NG_ONE2MANY_MAX_LINKS
}
```

Definition at line 100 of file ng_one2many.c.

7.127.3.8 `struct ng_parse_type ng_one2many_link_stats_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_one2many_link_stats_type_fields
}
```

Definition at line 118 of file ng_one2many.c.

7.127.3.9 `struct ng_parse_struct_field ng_one2many_link_stats_type_fields[] =`
`NG_ONE2MANY_LINK_STATS_TYPE_INFO` [static]

Definition at line 117 of file ng_one2many.c.

7.127.3.10 `ng_newhook_t ng_one2many_newhook` [static]

Definition at line 86 of file ng_one2many.c.

7.127.3.11 `ng_rcvdata_t ng_one2many_rcvdata` [static]

Definition at line 87 of file `ng_one2many.c`.

7.127.3.12 `ng_rcvmsg_t ng_one2many_rcvmsg` [static]

Definition at line 84 of file `ng_one2many.c`.

7.127.3.13 `ng_shutdown_t ng_one2many_shutdown` [static]

Definition at line 85 of file `ng_one2many.c`.

7.127.3.14 `struct ng_type ng_one2many_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_ONE2MANY_NODE_TYPE,
    .constructor = ng_one2many_constructor,
    .rcvmsg =      ng_one2many_rcvmsg,
    .shutdown =    ng_one2many_shutdown,
    .newhook =     ng_one2many_newhook,
    .rcvdata =     ng_one2many_rcvdata,
    .disconnect =  ng_one2many_disconnect,
    .cmdlist =     ng_one2many_cmdlist,
}
```

Definition at line 164 of file `ng_one2many.c`.

7.128 /usr/src/sys/netgraph/ng_one2many.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_one2many_config](#)
- struct [ng_one2many_link_stats](#)

Defines

- #define [NG_ONE2MANY_NODE_TYPE](#) "one2many"
- #define [NGM_ONE2MANY_COOKIE](#) 1100897444
- #define [NG_ONE2MANY_HOOK_ONE](#) "one"
- #define [NG_ONE2MANY_HOOK_MANY_PREFIX](#) "many"
- #define [NG_ONE2MANY_HOOK_MANY_FMT](#) "many%d"
- #define [NG_ONE2MANY_MAX_LINKS](#) 64
- #define [NG_ONE2MANY_ONE_LINKNUM](#) (-1)
- #define [NG_ONE2MANY_XMIT_ROUNDROBIN](#) 1
- #define [NG_ONE2MANY_XMIT_ALL](#) 2
- #define [NG_ONE2MANY_FAIL_MANUAL](#) 1
- #define [NG_ONE2MANY_FAIL_NOTIFY](#) 2
- #define [NG_ONE2MANY_CONFIG_TYPE_INFO](#)(atype)
- #define [NG_ONE2MANY_LINK_STATS_TYPE_INFO](#)

Enumerations

- enum {
 - [NGM_ONE2MANY_SET_CONFIG](#), [NGM_ONE2MANY_GET_CONFIG](#), [NGM_ONE2MANY_GET_STATS](#), [NGM_ONE2MANY_CLR_STATS](#), [NGM_ONE2MANY_GETCLR_STATS](#) }

7.128.1 Define Documentation

7.128.1.1 #define [NG_ONE2MANY_CONFIG_TYPE_INFO](#)(atype)

Value:

```

{
    \
    { "xmitAlg",          &ng_parse_uint32_type  }, \
    { "failAlg",         &ng_parse_uint32_type  }, \
    { "enabledLinks",   (atype)                }, \
    { NULL }
}

```

Definition at line 77 of file [ng_one2many.h](#).

7.128.1.2 #define NG_ONE2MANY_FAIL_MANUAL 1

Definition at line 66 of file ng_one2many.h.

Referenced by ng_one2many_constructor(), ng_one2many_rcvmsg(), and ng_one2many_update_many().

7.128.1.3 #define NG_ONE2MANY_FAIL_NOTIFY 2

Definition at line 67 of file ng_one2many.h.

Referenced by ng_one2many_rcvmsg(), and ng_one2many_update_many().

7.128.1.4 #define NG_ONE2MANY_HOOK_MANY_FMT "many%d"

Definition at line 53 of file ng_one2many.h.

7.128.1.5 #define NG_ONE2MANY_HOOK_MANY_PREFIX "many"

Definition at line 52 of file ng_one2many.h.

Referenced by ng_one2many_newhook().

7.128.1.6 #define NG_ONE2MANY_HOOK_ONE "one"

Definition at line 51 of file ng_one2many.h.

Referenced by ng_one2many_newhook().

7.128.1.7 #define NG_ONE2MANY_LINK_STATS_TYPE_INFO**Value:**

```
{
    \
    { "recvOctets",      &ng_parse_uint64_type }, \
    { "recvPackets",    &ng_parse_uint64_type }, \
    { "xmitOctets",     &ng_parse_uint64_type }, \
    { "xmitPackets",    &ng_parse_uint64_type }, \
    { "memoryFailures", &ng_parse_uint64_type }, \
    { NULL }
}
```

Definition at line 94 of file ng_one2many.h.

7.128.1.8 #define NG_ONE2MANY_MAX_LINKS 64

Definition at line 56 of file ng_one2many.h.

Referenced by ng_one2many_disconnect(), ng_one2many_newhook(), ng_one2many_rcvdata(), ng_one2many_rcvmsg(), and ng_one2many_update_many().

7.128.1.9 #define NG_ONE2MANY_NODE_TYPE "one2many"

Definition at line 47 of file ng_one2many.h.

7.128.1.10 #define NG_ONE2MANY_ONE_LINKNUM (-1)

Definition at line 59 of file ng_one2many.h.

Referenced by ng_one2many_disconnect(), ng_one2many_newhook(), ng_one2many_rcvdata(), and ng_one2many_rcvmsg().

7.128.1.11 #define NG_ONE2MANY_XMIT_ALL 2

Definition at line 63 of file ng_one2many.h.

Referenced by ng_one2many_rcvdata(), and ng_one2many_rcvmsg().

7.128.1.12 #define NG_ONE2MANY_XMIT_ROUNDROBIN 1

Definition at line 62 of file ng_one2many.h.

Referenced by ng_one2many_constructor(), ng_one2many_rcvdata(), and ng_one2many_rcvmsg().

7.128.1.13 #define NGM_ONE2MANY_COOKIE 1100897444

Definition at line 48 of file ng_one2many.h.

Referenced by ng_one2many_rcvmsg().

7.128.2 Enumeration Type Documentation**7.128.2.1 anonymous enum**

Enumerator:

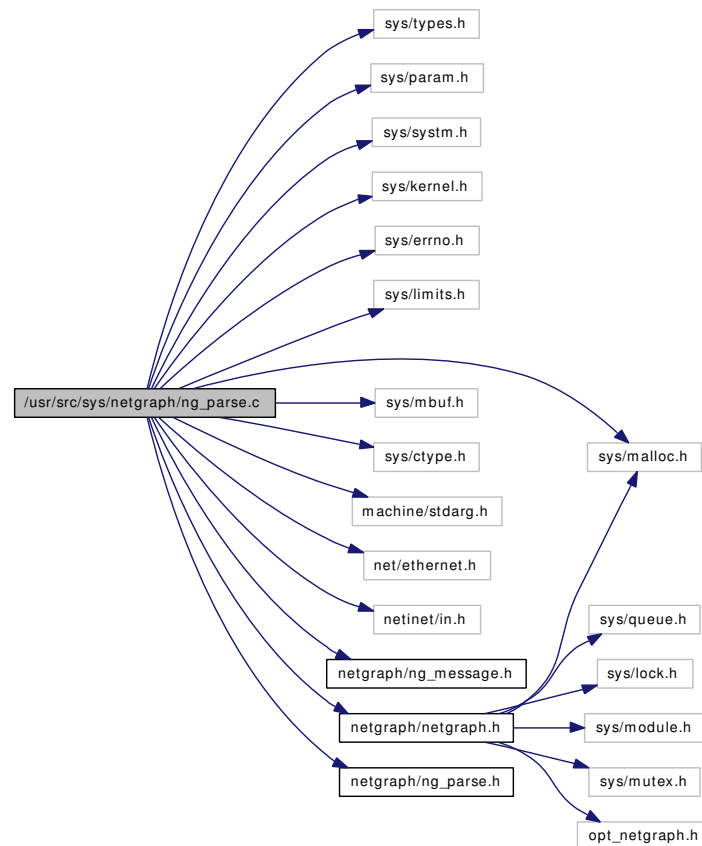
NGM_ONE2MANY_SET_CONFIG
NGM_ONE2MANY_GET_CONFIG
NGM_ONE2MANY_GET_STATS
NGM_ONE2MANY_CLR_STATS
NGM_ONE2MANY_GETCLR_STATS

Definition at line 104 of file ng_one2many.h.

7.129 /usr/src/sys/netgraph/ng_parse.c File Reference

```
#include <sys/types.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/errno.h>
#include <sys/limits.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/ctype.h>
#include <machine/stdarg.h>
#include <net/ethernet.h>
#include <netinet/in.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
```

Include dependency graph for ng_parse.c:



Data Structures

- struct [int16_temp](#)
- struct [int32_temp](#)
- struct [int64_temp](#)

Defines

- #define [M_NETGRAPH_PARSE](#) M_NETGRAPH
- #define [INT8_ALIGNMENT](#) 1
- #define [INT16_ALIGNMENT](#) ((int)&((struct [int16_temp](#) *)0) → y)
- #define [INT32_ALIGNMENT](#) ((int)&((struct [int32_temp](#) *)0) → y)
- #define [INT64_ALIGNMENT](#) ((int)&((struct [int64_temp](#) *)0) → y)
- #define [INT_UNSIGNED](#) 0
- #define [INT_SIGNED](#) 1
- #define [INT_HEX](#) 2
- #define [METHOD](#)(t, m) (ng_get_ ## m ## _method(t))
- #define [INVOKE](#)(t, m) (*METHOD(t,m))
- #define [ALIGNMENT](#)(t)

Enumerations

- enum [comptype](#) { [CT_STRUCT](#), [CT_ARRAY](#), [CT_FIXEDARRAY](#) }

Functions

- static int `ng_parse_composite` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *start, u_char *const buf, int *buflen, enum `comptype` ctype)
- static int `ng_unparse_composite` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen, enum `comptype` ctype)
- static int `ng_get_composite_elem_default` (const struct `ng_parse_type` *type, int index, const u_char *start, u_char *buf, int *buflen, enum `comptype` ctype)
- static int `ng_get_composite_len` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf, enum `comptype` ctype)
- static struct `ng_parse_type` * `ng_get_composite_etype` (const struct `ng_parse_type` *type, int index, enum `comptype` ctype)
- static int `ng_parse_get_elem_pad` (const struct `ng_parse_type` *type, int index, enum `comptype` ctype, int posn)
- static int `ng_parse_skip_value` (const char *s, int off, int *lenp)
- static int `ng_parse_append` (char **cbufp, int *cbuflenp, const char *fmt,...)
- static `ng_parse_t` * `ng_get_parse_method` (const struct `ng_parse_type` *t)
- static `ng_unparse_t` * `ng_get_unparse_method` (const struct `ng_parse_type` *t)
- static `ng_getDefault_t` * `ng_get_getDefault_method` (const struct `ng_parse_type` *t)
- static `ng_getAlign_t` * `ng_get_getAlign_method` (const struct `ng_parse_type` *t)
- int `ng_parse` (const struct `ng_parse_type` *type, const char *string, int *off, u_char *buf, int *buflen)
- int `ng_unparse` (const struct `ng_parse_type` *type, const u_char *data, char *cbuf, int cbuflen)
- int `ng_parse_getDefault` (const struct `ng_parse_type` *type, u_char *buf, int *buflen)
- static int `ng_struct_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_struct_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_struct_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_struct_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_fixedarray_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_fixedarray_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_fixedarray_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_ - char *buf, int *buflen)
- static int `ng_fixedarray_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_array_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_array_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_array_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_array_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_int8_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_int8_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_int8_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_int8_getAlign` (const struct `ng_parse_type` *type)

- static int `ng_int16_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_int16_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_int16_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_int16_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_int32_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_int32_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_int32_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_int32_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_int64_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_int64_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_int64_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_int64_getAlign` (const struct `ng_parse_type` *type)
- static int `ng_string_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_string_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_string_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_fixedstring_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_fixedstring_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_fixedstring_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_sizedstring_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_sizedstring_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_sizedstring_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_ipaddr_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_ipaddr_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_ipaddr_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_enaddr_parse` (const struct `ng_parse_type` *type, const char *s, int *const off, const u_char *const start, u_char *const buf, int *const buflen)
- static int `ng_enaddr_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_parse_bytearray_subtype_getLength` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf)

- static int `ng_bytearray_parse` (const struct `ng_parse_type` *type, const char *s, int *off, const u_char *const start, u_char *const buf, int *buflen)
- static int `ng_bytearray_unparse` (const struct `ng_parse_type` *type, const u_char *data, int *off, char *cbuf, int cbuflen)
- static int `ng_bytearray_getDefault` (const struct `ng_parse_type` *type, const u_char *const start, u_char *buf, int *buflen)
- static int `ng_parse_ng_mesg_getLength` (const struct `ng_parse_type` *type, const u_char *start, const u_char *buf)
- enum `ng_parse_token` `ng_parse_get_token` (const char *s, int *startp, int *lenp)
- char * `ng_get_string_token` (const char *s, int *startp, int *lenp, int *slenp)
- char * `ng_encode_string` (const char *raw, int slen)

Variables

- `ng_parse_type` `ng_parse_struct_type`
- `ng_parse_type` `ng_parse_fixedarray_type`
- `ng_parse_type` `ng_parse_array_type`
- `ng_parse_type` `ng_parse_int8_type`
- `ng_parse_type` `ng_parse_uint8_type`
- `ng_parse_type` `ng_parse_hint8_type`
- `ng_parse_type` `ng_parse_int16_type`
- `ng_parse_type` `ng_parse_uint16_type`
- `ng_parse_type` `ng_parse_hint16_type`
- `ng_parse_type` `ng_parse_int32_type`
- `ng_parse_type` `ng_parse_uint32_type`
- `ng_parse_type` `ng_parse_hint32_type`
- `ng_parse_type` `ng_parse_int64_type`
- `ng_parse_type` `ng_parse_uint64_type`
- `ng_parse_type` `ng_parse_hint64_type`
- `ng_parse_type` `ng_parse_string_type`
- `ng_parse_type` `ng_parse_fixedstring_type`
- `ng_parse_fixedstring_info` `ng_parse_nodebuf_info`
- `ng_parse_type` `ng_parse_nodebuf_type`
- `ng_parse_fixedstring_info` `ng_parse_hookbuf_info`
- `ng_parse_type` `ng_parse_hookbuf_type`
- `ng_parse_fixedstring_info` `ng_parse_pathbuf_info`
- `ng_parse_type` `ng_parse_pathbuf_type`
- `ng_parse_fixedstring_info` `ng_parse_typebuf_info`
- `ng_parse_type` `ng_parse_typebuf_type`
- `ng_parse_fixedstring_info` `ng_parse_cmdbuf_info`
- `ng_parse_type` `ng_parse_cmdbuf_type`
- `ng_parse_type` `ng_parse_sizedstring_type`
- `ng_parse_type` `ng_parse_ipaddr_type`
- `ng_parse_type` `ng_parse_enaddr_type`
- static struct `ng_parse_array_info` `ng_parse_bytearray_subtype_info`
- static struct `ng_parse_type` `ng_parse_bytearray_subtype`
- `ng_parse_type` `ng_parse_bytearray_type`
- static struct `ng_parse_type` `ng_msg_data_type`
- static struct `ng_parse_struct_field` `ng_parse_ng_mesg_type_fields` [] = `NG_GENERIC_NG_MESG_INFO(&ng_msg_data_type)`
- `ng_parse_type` `ng_parse_ng_mesg_type`

7.129.1 Define Documentation

7.129.1.1 #define ALIGNMENT(t)

Value:

```
(METHOD(t, getAlign) == NULL ? \  
0 : INVOKE(t, getAlign)(t))
```

Definition at line 136 of file ng_parse.c.

Referenced by ng_array_getAlign(), ng_fixedarray_getAlign(), ng_parse_composite(), ng_parse_get_elem_pad(), and ng_struct_getAlign().

7.129.1.2 #define INT16_ALIGNMENT ((int)&((struct int16_temp *)0) → y)

Definition at line 87 of file ng_parse.c.

Referenced by ng_int16_getAlign().

7.129.1.3 #define INT32_ALIGNMENT ((int)&((struct int32_temp *)0) → y)

Definition at line 88 of file ng_parse.c.

Referenced by ng_int32_getAlign().

7.129.1.4 #define INT64_ALIGNMENT ((int)&((struct int64_temp *)0) → y)

Definition at line 89 of file ng_parse.c.

Referenced by ng_int64_getAlign().

7.129.1.5 #define INT8_ALIGNMENT 1

Definition at line 86 of file ng_parse.c.

Referenced by ng_int8_getAlign().

7.129.1.6 #define INT_HEX 2

Definition at line 94 of file ng_parse.c.

Referenced by ng_int16_unparse(), ng_int32_unparse(), ng_int64_unparse(), and ng_int8_unparse().

7.129.1.7 #define INT_SIGNED 1

Definition at line 93 of file ng_parse.c.

Referenced by ng_int16_unparse(), ng_int32_parse(), ng_int32_unparse(), ng_int64_unparse(), and ng_int8_unparse().

7.129.1.8 #define INT_UNSIGNED 0

Definition at line 92 of file ng_parse.c.

Referenced by ng_int16_unparse(), ng_int32_unparse(), ng_int64_unparse(), and ng_int8_unparse().

7.129.1.9 #define INVOKE(t, m) (*METHOD(t,m))

Definition at line 128 of file ng_parse.c.

Referenced by ng_parse(), ng_parse_composite(), ng_unparse(), and ng_unparse_composite().

7.129.1.10 #define M_NETGRAPH_PARSE M_NETGRAPH

Definition at line 67 of file ng_parse.c.

Referenced by ng_bytearray_parse(), ng_encode_string(), ng_fixedstring_parse(), ng_get_string_token(), ng_parse_composite(), ng_parse_get_token(), ng_sizedstring_parse(), ng_sizedstring_unparse(), ng_string_parse(), ng_string_unparse(), and ng_unparse_composite().

7.129.1.11 #define METHOD(t, m) (ng_get_ ## m ## _method(t))

Definition at line 127 of file ng_parse.c.

Referenced by ng_get_composite_elem_default(), and ng_parse_getDefault().

7.129.2 Enumeration Type Documentation**7.129.2.1 enum [comptype](#)**

Enumerator:

CT_STRUCT

CT_ARRAY

CT_FIXEDARRAY

Definition at line 97 of file ng_parse.c.

7.129.3 Function Documentation**7.129.3.1 static int ng_array_getAlign (const struct [ng_parse_type](#) * type) [static]**

Definition at line 312 of file ng_parse.c.

References ALIGNMENT, ng_parse_array_info::elementType, and ng_parse_type::info.

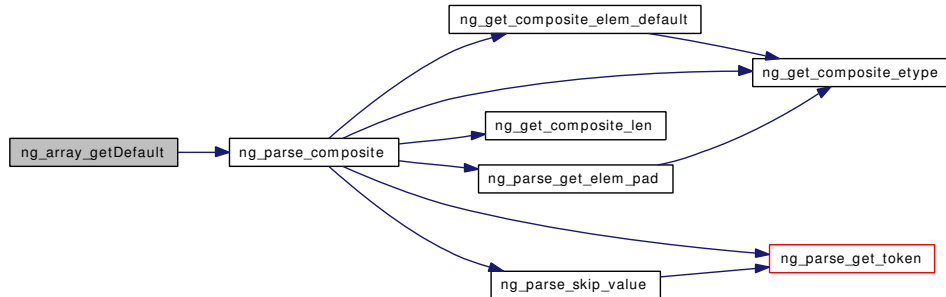
7.129.3.2 static int ng_array_getDefault (const struct [ng_parse_type](#) * type, const u_char *const start, u_char * buf, int * buflen) [static]

Definition at line 302 of file ng_parse.c.

References CT_ARRAY, and ng_parse_composite().

Referenced by `ng_bytearray_getDefault()`.

Here is the call graph for this function:



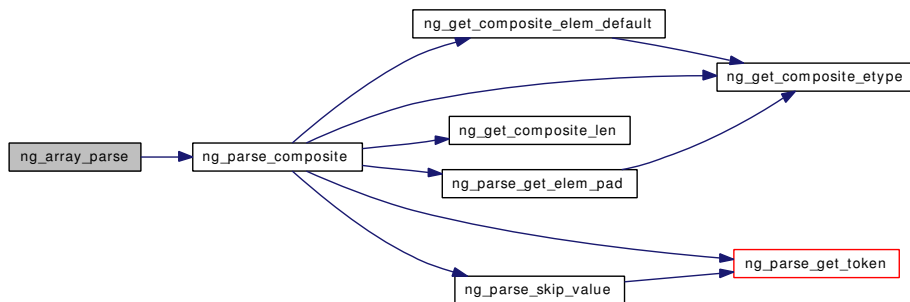
7.129.3.3 `static int ng_array_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen) [static]`

Definition at line 287 of file `ng_parse.c`.

References `CT_ARRAY`, and `ng_parse_composite()`.

Referenced by `ng_bytearray_parse()`.

Here is the call graph for this function:



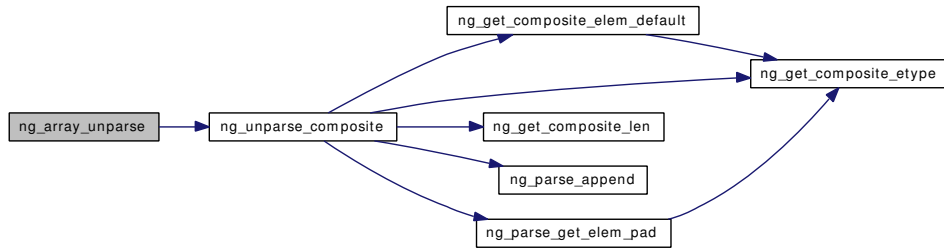
7.129.3.4 `static int ng_array_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen) [static]`

Definition at line 295 of file `ng_parse.c`.

References `CT_ARRAY`, and `ng_unparse_composite()`.

Referenced by `ng_bytearray_unparse()`.

Here is the call graph for this function:

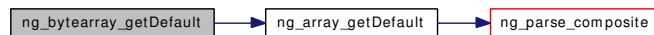


7.129.3.5 `static int ng_bytearray_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 1154 of file ng_parse.c.

References `ng_parse_type::info`, `ng_array_getDefault()`, `ng_parse_bytearray_subtype`, and `ng_parse_type::private`.

Here is the call graph for this function:

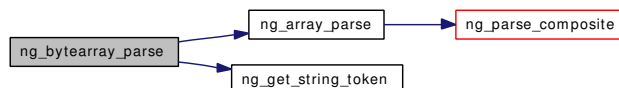


7.129.3.6 `static int ng_bytearray_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 1105 of file ng_parse.c.

References `ng_parse_type::info`, `M_NETGRAPH_PARSE`, `ng_array_parse()`, `ng_get_string_token()`, `ng_parse_bytearray_subtype`, and `ng_parse_type::private`.

Here is the call graph for this function:

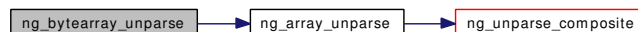


7.129.3.7 `static int ng_bytearray_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 1143 of file ng_parse.c.

References `ng_parse_type::info`, `ng_array_unparse()`, `ng_parse_bytearray_subtype`, and `ng_parse_type::private`.

Here is the call graph for this function:



7.129.3.8 `static int ng_enaddr_parse (const struct ng_parse_type * type, const char * s, int *const off, const u_char *const start, u_char *const buf, int *const buflen) [static]`

Definition at line 1028 of file `ng_parse.c`.

7.129.3.9 `static int ng_enaddr_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen) [static]`

Definition at line 1055 of file `ng_parse.c`.

7.129.3.10 `char* ng_encode_string (const char * raw, int slen)`

Definition at line 1825 of file `ng_parse.c`.

References `M_NETGRAPH_PARSE`.

Referenced by `ng_ksocket_sockaddr_unparse()`, `ng_sizedstring_unparse()`, and `ng_string_unparse()`.

7.129.3.11 `static int ng_fixedarray_getAlign (const struct ng_parse_type * type) [static]`

Definition at line 265 of file `ng_parse.c`.

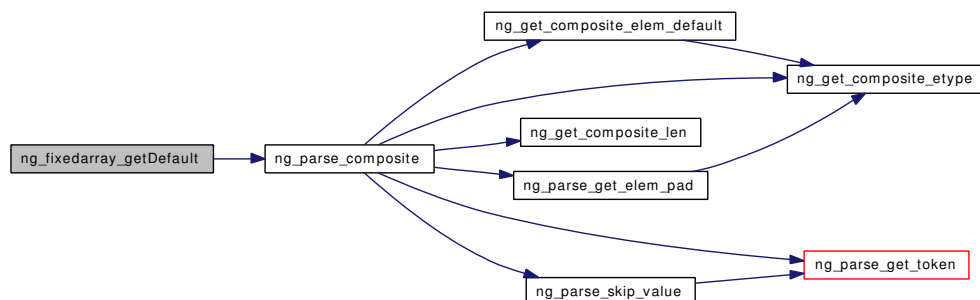
References `ALIGNMENT`, `ng_parse_fixedarray_info::elementType`, and `ng_parse_type::info`.

7.129.3.12 `static int ng_fixedarray_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen) [static]`

Definition at line 255 of file `ng_parse.c`.

References `CT_FIXEDARRAY`, and `ng_parse_composite()`.

Here is the call graph for this function:

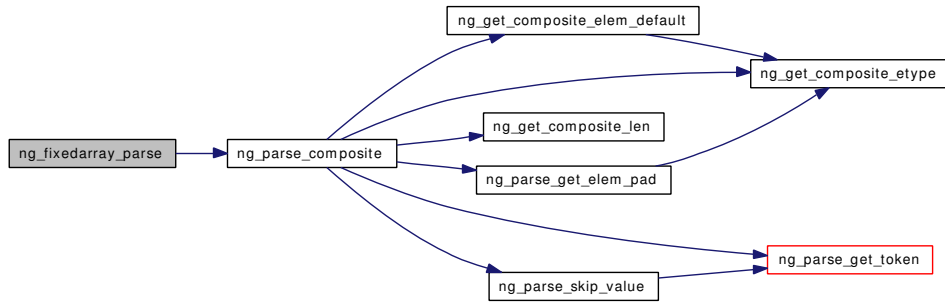


7.129.3.13 `static int ng_fixedarray_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen) [static]`

Definition at line 238 of file `ng_parse.c`.

References `CT_FIXEDARRAY`, and `ng_parse_composite()`.

Here is the call graph for this function:

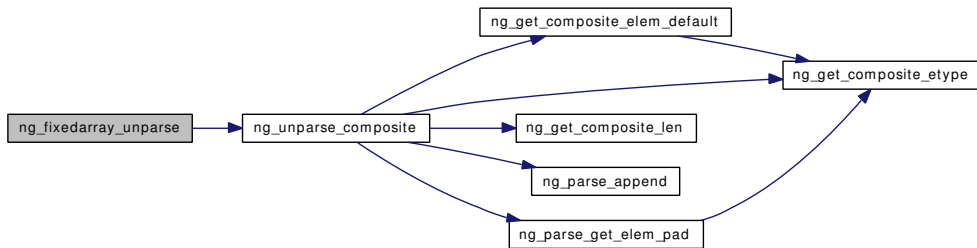


7.129.3.14 `static int ng_fixedarray_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 247 of file `ng_parse.c`.

References `CT_FIXEDARRAY`, and `ng_unparse_composite()`.

Here is the call graph for this function:



7.129.3.15 `static int ng_fixedstring_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 831 of file `ng_parse.c`.

References `ng_parse_fixedstring_info::bufSize`, and `ng_parse_type::info`.

7.129.3.16 `static int ng_fixedstring_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 794 of file `ng_parse.c`.

References `ng_parse_fixedstring_info::bufSize`, `ng_parse_type::info`, `M_NETGRAPH_PARSE`, and `ng_get_string_token()`.

Here is the call graph for this function:

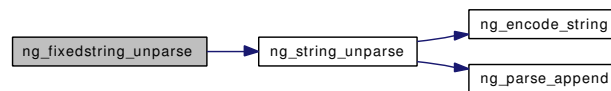


7.129.3.17 `static int ng_fixedstring_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 818 of file ng_parse.c.

References `ng_parse_fixedstring_info::bufSize`, `ng_parse_type::info`, and `ng_string_unparse()`.

Here is the call graph for this function:



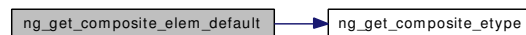
7.129.3.18 `static int ng_get_composite_elem_default (const struct ng_parse_type * type, int index, const u_char * start, u_char * buf, int * buflen, enum comptype ctype)` [static]

Definition at line 1479 of file ng_parse.c.

References `CT_ARRAY`, `CT_FIXEDARRAY`, `CT_STRUCT`, `ng_parse_fixedarray_info::getDefault`, `ng_parse_array_info::getDefault`, `ng_parse_type::info`, `METHOD`, and `ng_get_composite_etype()`.

Referenced by `ng_parse_composite()`, and `ng_unparse_composite()`.

Here is the call graph for this function:



7.129.3.19 `static struct ng_parse_type * ng_get_composite_etype (const struct ng_parse_type * type, int index, enum comptype ctype)` [static]

Definition at line 1566 of file ng_parse.c.

References `CT_ARRAY`, `CT_FIXEDARRAY`, `CT_STRUCT`, `ng_parse_fixedarray_info::elementType`, `ng_parse_array_info::elementType`, `ng_parse_type::info`, and `ng_parse_struct_field::type`.

Referenced by `ng_get_composite_elem_default()`, `ng_parse_composite()`, `ng_parse_get_elem_pad()`, and `ng_unparse_composite()`.

7.129.3.20 `static int ng_get_composite_len (const struct ng_parse_type * type, const u_char * start, const u_char * buf, enum comptype ctype)` [static]

Definition at line 1525 of file ng_parse.c.

References `CT_ARRAY`, `CT_FIXEDARRAY`, `CT_STRUCT`, `ng_parse_array_info::getLength`, `ng_parse_type::info`, `ng_parse_fixedarray_info::length`, and `ng_parse_struct_field::type`.

Referenced by `ng_parse_composite()`, and `ng_unparse_composite()`.

7.129.3.21 `static ng_getAlign_t * ng_get_getAlign_method (const struct ng_parse_type * t)` [static]

Definition at line 1906 of file ng_parse.c.

References `ng_parse_type::getAlign`, and `ng_parse_type::supertype`.

7.129.3.22 `static ng_getDefault_t * ng_get_getDefault_method (const struct ng_parse_type * t)`
[static]

Definition at line 1898 of file `ng_parse.c`.

References `ng_parse_type::getDefault`, and `ng_parse_type::supertype`.

7.129.3.23 `static ng_parse_t * ng_get_parse_method (const struct ng_parse_type * t)` [static]

Definition at line 1882 of file `ng_parse.c`.

References `ng_parse_type::parse`, and `ng_parse_type::supertype`.

7.129.3.24 `char* ng_get_string_token (const char * s, int * startp, int * lenp, int * slenp)`

Definition at line 1737 of file `ng_parse.c`.

References `M_NETGRAPH_PARSE`.

Referenced by `ng_bytearray_parse()`, `ng_fixedstring_parse()`, `ng_ksocket_sockaddr_parse()`, `ng_parse_get_token()`, `ng_sizedstring_parse()`, and `ng_string_parse()`.

7.129.3.25 `static ng_unparse_t * ng_get_unparse_method (const struct ng_parse_type * t)`
[static]

Definition at line 1890 of file `ng_parse.c`.

References `ng_parse_type::supertype`, and `ng_parse_type::unparse`.

7.129.3.26 `static int ng_int16_getAlign (const struct ng_parse_type * type)` [static]

Definition at line 501 of file `ng_parse.c`.

References `INT16_ALIGNMENT`.

7.129.3.27 `static int ng_int16_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 487 of file `ng_parse.c`.

7.129.3.28 `static int ng_int16_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 432 of file `ng_parse.c`.

7.129.3.29 `static int ng_int16_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 452 of file `ng_parse.c`.

References `ng_parse_type::info`, `INT_HEX`, `INT_SIGNED`, `INT_UNSIGNED`, and `ng_parse_append()`.

Here is the call graph for this function:



7.129.3.30 `static int ng_int32_getAlign (const struct ng_parse_type * type)` [static]

Definition at line 603 of file `ng_parse.c`.

References `INT32_ALIGNMENT`.

7.129.3.31 `static int ng_int32_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 589 of file `ng_parse.c`.

7.129.3.32 `static int ng_int32_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 531 of file `ng_parse.c`.

References `ng_parse_type::info`, and `INT_SIGNED`.

7.129.3.33 `static int ng_int32_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 554 of file `ng_parse.c`.

References `ng_parse_type::info`, `INT_HEX`, `INT_SIGNED`, `INT_UNSIGNED`, and `ng_parse_append()`.

Here is the call graph for this function:



7.129.3.34 `static int ng_int64_getAlign (const struct ng_parse_type * type)` [static]

Definition at line 701 of file `ng_parse.c`.

References `INT64_ALIGNMENT`.

7.129.3.35 `static int ng_int64_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 687 of file `ng_parse.c`.

7.129.3.36 `static int ng_int64_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 633 of file `ng_parse.c`.

7.129.3.37 `static int ng_int64_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 652 of file `ng_parse.c`.

References `ng_parse_type::info`, `INT_HEX`, `INT_SIGNED`, `INT_UNSIGNED`, and `ng_parse_append()`.

Here is the call graph for this function:



7.129.3.38 `static int ng_int8_getAlign (const struct ng_parse_type * type)` [static]

Definition at line 402 of file `ng_parse.c`.

References `INT8_ALIGNMENT`.

7.129.3.39 `static int ng_int8_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 388 of file `ng_parse.c`.

7.129.3.40 `static int ng_int8_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 334 of file `ng_parse.c`.

Referenced by `ng_ipaddr_parse()`.

7.129.3.41 `static int ng_int8_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 353 of file `ng_parse.c`.

References `ng_parse_type::info`, `INT_HEX`, `INT_SIGNED`, `INT_UNSIGNED`, and `ng_parse_append()`.

Here is the call graph for this function:



7.129.3.42 `static int ng_ipaddr_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 1001 of file ng_parse.c.

7.129.3.43 `static int ng_ipaddr_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 966 of file ng_parse.c.

References `ng_int8_parse()`, and `ng_parse_int8_type`.

Here is the call graph for this function:



7.129.3.44 `static int ng_ipaddr_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 985 of file ng_parse.c.

References `ng_parse_append()`.

Here is the call graph for this function:



7.129.3.45 `int ng_parse (const struct ng_parse_type * type, const char * string, int * off, u_char * buf, int * buflen)`

Definition at line 147 of file ng_parse.c.

References `INVOKE`, and `ng_parse_type::parse`.

Referenced by `ng_generic_msg()`.

7.129.3.46 `static int ng_parse_append (char ** cbuftp, int * cbuflenp, const char * fmt, ...)` [static]

Definition at line 1632 of file ng_parse.c.

Referenced by `ng_int16_unparse()`, `ng_int32_unparse()`, `ng_int64_unparse()`, `ng_int8_unparse()`, `ng_ipaddr_unparse()`, `ng_sizedstring_unparse()`, `ng_string_unparse()`, and `ng_unparse_composite()`.

7.129.3.47 `static int ng_parse_bytearray_subtype_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 1085 of file ng_parse.c.

References `ng_parse_type::private`.

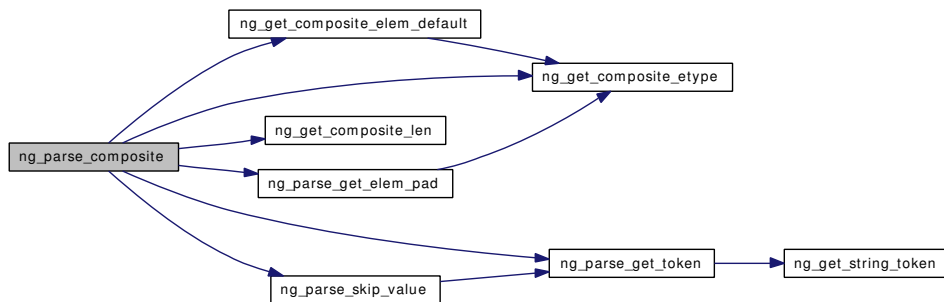
7.129.3.48 `static int ng_parse_composite (const struct ng_parse_type * type, const char * s, int * off, const u_char * start, u_char * const buf, int * buflen, enum comptype ctype)`
 [static]

Definition at line 1211 of file ng_parse.c.

References ALIGNMENT, CT_STRUCT, ng_parse_type::info, INVOKE, M_NETGRAPH_PARSE, ng_get_composite_elem_default(), ng_get_composite_etype(), ng_get_composite_len(), ng_parse_get_elem_pad(), ng_parse_get_token(), ng_parse_skip_value(), T_EQUALS, T_LBRACE, T_LBRACKET, T_RBRACE, T_RBRACKET, T_WORD, and ng_parse_struct_field::type.

Referenced by ng_array_getDefault(), ng_array_parse(), ng_fixedarray_getDefault(), ng_fixedarray_parse(), ng_struct_getDefault(), and ng_struct_parse().

Here is the call graph for this function:



7.129.3.49 `static int ng_parse_get_elem_pad (const struct ng_parse_type * type, int index, enum comptype ctype, int posn)` [static]

Definition at line 1604 of file ng_parse.c.

References ng_parse_struct_field::alignment, ALIGNMENT, CT_STRUCT, ng_parse_type::info, ng_get_composite_etype(), and ng_parse_struct_field::type.

Referenced by ng_parse_composite(), and ng_unparse_composite().

Here is the call graph for this function:



7.129.3.50 `enum ng_parse_token ng_parse_get_token (const char * s, int * startp, int * lenp)`

Definition at line 1691 of file ng_parse.c.

References M_NETGRAPH_PARSE, ng_get_string_token(), T_EOF, T_EQUALS, T_ERROR, T_LBRACE, T_LBRACKET, T_RBRACE, T_RBRACKET, T_STRING, and T_WORD.

Referenced by ng_ksocket_sockaddr_parse(), ng_parse_composite(), and ng_parse_skip_value().

Here is the call graph for this function:



7.129.3.51 `int ng_parse_getDefault (const struct ng_parse_type * type, u_char * buf, int * buflen)`

Definition at line 169 of file `ng_parse.c`.

References `ng_parse_type::getDefault`, and `METHOD`.

7.129.3.52 `static int ng_parse_ng_mesg_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf)` [static]

Definition at line 1180 of file `ng_parse.c`.

References `ng_mesg::ng_msghdr::arglen`, and `ng_mesg::header`.

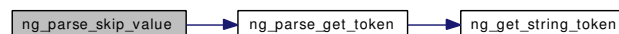
7.129.3.53 `static int ng_parse_skip_value (const char * s, int off, int * lenp)` [static]

Definition at line 1652 of file `ng_parse.c`.

References `ng_parse_get_token()`, `T_EOF`, `T_LBRACE`, `T_LBRACKET`, `T_RBRACE`, and `T_RBRACKET`.

Referenced by `ng_parse_composite()`.

Here is the call graph for this function:



7.129.3.54 `static int ng_sizedstring_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 941 of file `ng_parse.c`.

7.129.3.55 `static int ng_sizedstring_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 898 of file `ng_parse.c`.

References `M_NETGRAPH_PARSE`, and `ng_get_string_token()`.

Here is the call graph for this function:

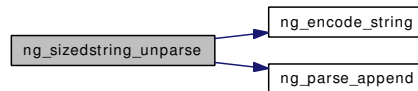


7.129.3.56 `static int ng_sizedstring_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 921 of file ng_parse.c.

References M_NETGRAPH_PARSE, ng_encode_string(), and ng_parse_append().

Here is the call graph for this function:



7.129.3.57 `static int ng_string_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen)` [static]

Definition at line 768 of file ng_parse.c.

7.129.3.58 `static int ng_string_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen)` [static]

Definition at line 731 of file ng_parse.c.

References M_NETGRAPH_PARSE, and ng_get_string_token().

Here is the call graph for this function:



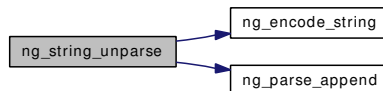
7.129.3.59 `static int ng_string_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen)` [static]

Definition at line 749 of file ng_parse.c.

References M_NETGRAPH_PARSE, ng_encode_string(), and ng_parse_append().

Referenced by ng_fixedstring_unparse().

Here is the call graph for this function:



7.129.3.60 `static int ng_struct_getAlign (const struct ng_parse_type * type)` [static]

Definition at line 209 of file ng_parse.c.

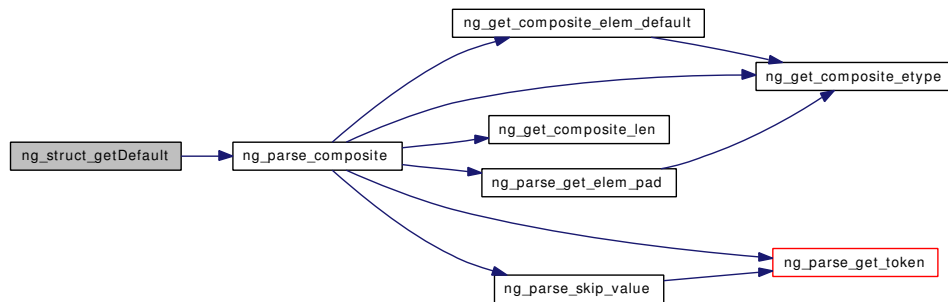
References ALIGNMENT, ng_parse_type::info, ng_parse_struct_field::name, and ng_parse_struct_field::type.

7.129.3.61 static int ng_struct_getDefault (const struct ng_parse_type * type, const u_char *const start, u_char * buf, int * buflen) [static]

Definition at line 199 of file ng_parse.c.

References CT_STRUCT, and ng_parse_composite().

Here is the call graph for this function:

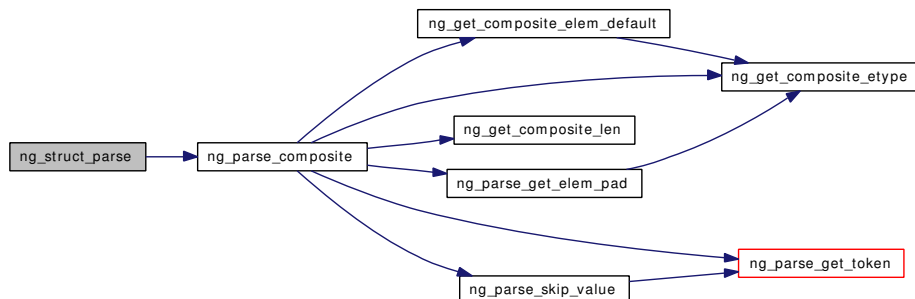


7.129.3.62 static int ng_struct_parse (const struct ng_parse_type * type, const char * s, int * off, const u_char *const start, u_char *const buf, int * buflen) [static]

Definition at line 184 of file ng_parse.c.

References CT_STRUCT, and ng_parse_composite().

Here is the call graph for this function:

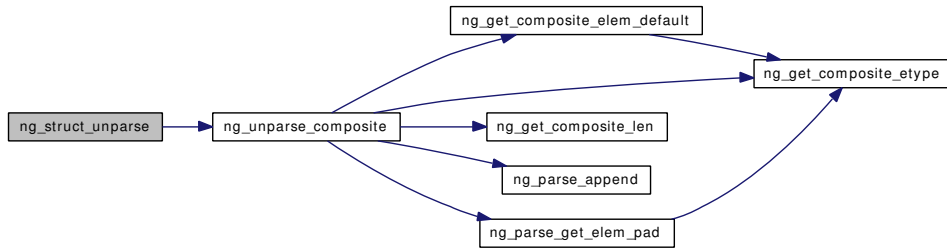


7.129.3.63 static int ng_struct_unparse (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen) [static]

Definition at line 192 of file ng_parse.c.

References CT_STRUCT, and ng_unparse_composite().

Here is the call graph for this function:



7.129.3.64 `int ng_unparse (const struct ng_parse_type * type, const u_char * data, char * cbuf, int cbuflen)`

Definition at line 157 of file `ng_parse.c`.

References `INVOKE`, and `ng_parse_type::unparse`.

Referenced by `ng_generic_msg()`.

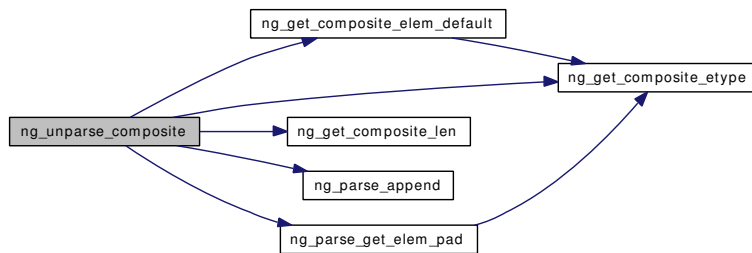
7.129.3.65 `static int ng_unparse_composite (const struct ng_parse_type * type, const u_char * data, int * off, char * cbuf, int cbuflen, enum comptype ctype)` `[static]`

Definition at line 1386 of file `ng_parse.c`.

References `CT_STRUCT`, `ng_parse_type::info`, `INVOKE`, `M_NETGRAPH_PARSE`, `name`, `ng_get_composite_elem_default()`, `ng_get_composite_etype()`, `ng_get_composite_len()`, `ng_parse_append()`, `ng_parse_get_elem_pad()`, and `ng_parse_struct_field::type`.

Referenced by `ng_array_unparse()`, `ng_fixedarray_unparse()`, and `ng_struct_unparse()`.

Here is the call graph for this function:



7.129.4 Variable Documentation

7.129.4.1 `struct ng_parse_type ng_msg_data_type` `[static]`

Initial value:

```

{
    &ng_parse_bytearray_type,
    &ng_parse_ng_msg_getLength
}

```

Definition at line 1190 of file `ng_parse.c`.

7.129.4.2 struct `ng_parse_type ng_parse_array_type`

Initial value:

```
{
    NULL,
    NULL,
    NULL,
    ng_array_parse,
    ng_array_unparse,
    ng_array_getDefault,
    ng_array_getAlign
}
```

Definition at line 319 of file `ng_parse.c`.

7.129.4.3 struct `ng_parse_type ng_parse_bytearray_subtype` [static]

Initial value:

```
{
    &ng_parse_array_type,
    &ng_parse_bytearray_subtype_info
}
```

Definition at line 1099 of file `ng_parse.c`.

Referenced by `ng_bytearray_getDefault()`, `ng_bytearray_parse()`, and `ng_bytearray_unparse()`.

7.129.4.4 struct `ng_parse_array_info ng_parse_bytearray_subtype_info` [static]

Initial value:

```
{
    &ng_parse_hint8_type,
    &ng_parse_bytearray_subtype_getLength,
    NULL
}
```

Definition at line 1094 of file `ng_parse.c`.

7.129.4.5 struct `ng_parse_type ng_parse_bytearray_type`

Initial value:

```
{
    NULL,
    NULL,
    NULL,
    ng_bytearray_parse,
    ng_bytearray_unparse,
    ng_bytearray_getDefault,
    NULL
}
```

Definition at line 1164 of file `ng_parse.c`.

7.129.4.6 struct [ng_parse_fixedstring_info](#) [ng_parse_cmdbuf_info](#)**Initial value:**

```
{
    NG_CMDSTRSIZ
}
```

Definition at line 885 of file `ng_parse.c`.

7.129.4.7 struct [ng_parse_type](#) [ng_parse_cmdbuf_type](#)**Initial value:**

```
{
    &ng_parse_fixedstring_type,
    &ng_parse_cmdbuf_info
}
```

Definition at line 888 of file `ng_parse.c`.

7.129.4.8 struct [ng_parse_type](#) [ng_parse_enaddr_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_enaddr_parse,
    ng_enaddr_unparse,
    NULL,
    0
}
```

Definition at line 1069 of file `ng_parse.c`.

7.129.4.9 struct [ng_parse_type](#) [ng_parse_fixedarray_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_fixedarray_parse,
    ng_fixedarray_unparse,
    ng_fixedarray_getDefault,
    ng_fixedarray_getAlign
}
```

Definition at line 272 of file `ng_parse.c`.

7.129.4.10 struct [ng_parse_type](#) [ng_parse_fixedstring_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_fixedstring_parse,
    ng_fixedstring_unparse,
    ng_fixedstring_getDefault,
    NULL
}
```

Definition at line 843 of file ng_parse.c.

7.129.4.11 struct [ng_parse_type](#) [ng_parse_hint16_type](#)**Initial value:**

```
{
    &ng_parse_int16_type,
    (void *)INT_HEX
}
```

Definition at line 521 of file ng_parse.c.

7.129.4.12 struct [ng_parse_type](#) [ng_parse_hint32_type](#)**Initial value:**

```
{
    &ng_parse_int32_type,
    (void *)INT_HEX
}
```

Definition at line 623 of file ng_parse.c.

7.129.4.13 struct [ng_parse_type](#) [ng_parse_hint64_type](#)**Initial value:**

```
{
    &ng_parse_int64_type,
    (void *)INT_HEX
}
```

Definition at line 721 of file ng_parse.c.

7.129.4.14 struct [ng_parse_type](#) [ng_parse_hint8_type](#)**Initial value:**

```
{
    &ng_parse_int8_type,
    (void *)INT_HEX
}
```

Definition at line 422 of file ng_parse.c.

7.129.4.15 struct [ng_parse_fixedstring_info](#) [ng_parse_hookbuf_info](#)

Initial value:

```
{
    NG_HOOKSIZ
}
```

Definition at line 861 of file ng_parse.c.

7.129.4.16 struct [ng_parse_type](#) [ng_parse_hookbuf_type](#)

Initial value:

```
{
    &ng_parse_fixedstring_type,
    &ng_parse_hookbuf_info
}
```

Definition at line 864 of file ng_parse.c.

7.129.4.17 struct [ng_parse_type](#) [ng_parse_int16_type](#)

Initial value:

```
{
    NULL,
    (void *)INT_SIGNED,
    NULL,
    ng_int16_parse,
    ng_int16_unparse,
    ng_int16_getDefault,
    ng_int16_getAlign
}
```

Definition at line 506 of file ng_parse.c.

7.129.4.18 struct [ng_parse_type](#) [ng_parse_int32_type](#)

Initial value:

```
{
    NULL,
    (void *)INT_SIGNED,
    NULL,
    ng_int32_parse,
    ng_int32_unparse,
    ng_int32_getDefault,
    ng_int32_getAlign
}
```


Definition at line 608 of file ng_parse.c.

7.129.4.19 struct [ng_parse_type](#) [ng_parse_int64_type](#)

Initial value:

```
{
    NULL,
    (void *)INT_SIGNED,
    NULL,
    ng_int64_parse,
    ng_int64_unparse,
    ng_int64_getDefault,
    ng_int64_getAlign
}
```

Definition at line 706 of file ng_parse.c.

7.129.4.20 struct [ng_parse_type](#) [ng_parse_int8_type](#)

Initial value:

```
{
    NULL,
    (void *)INT_SIGNED,
    NULL,
    ng_int8_parse,
    ng_int8_unparse,
    ng_int8_getDefault,
    ng_int8_getAlign
}
```

Definition at line 407 of file ng_parse.c.

Referenced by [ng_ipaddr_parse\(\)](#).

7.129.4.21 struct [ng_parse_type](#) [ng_parse_ipaddr_type](#)

Initial value:

```
{
    NULL,
    NULL,
    NULL,
    ng_ipaddr_parse,
    ng_ipaddr_unparse,
    ng_ipaddr_getDefault,
    ng_int32_getAlign
}
```

Definition at line 1013 of file ng_parse.c.

7.129.4.22 struct [ng_parse_type](#) [ng_parse_ng_mesg_type](#)

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_parse_ng_mesg_type_fields,
}
```

Definition at line 1198 of file ng_parse.c.

7.129.4.23 `struct ng_parse_struct_field ng_parse_ng_mesg_type_fields[] = NG_GENERIC_NG_MESG_INFO(&ng_msg_data_type) [static]`

Definition at line 1197 of file ng_parse.c.

7.129.4.24 `struct ng_parse_fixedstring_info ng_parse_nodebuf_info`

Initial value:

```
{
    NG_NODESIZ
}
```

Definition at line 853 of file ng_parse.c.

7.129.4.25 `struct ng_parse_type ng_parse_nodebuf_type`

Initial value:

```
{
    &ng_parse_fixedstring_type,
    &ng_parse_nodebuf_info
}
```

Definition at line 856 of file ng_parse.c.

7.129.4.26 `struct ng_parse_fixedstring_info ng_parse_pathbuf_info`

Initial value:

```
{
    NG_PATHSIZ
}
```

Definition at line 869 of file ng_parse.c.

7.129.4.27 `struct ng_parse_type ng_parse_pathbuf_type`

Initial value:

```
{
    &ng_parse_fixedstring_type,
    &ng_parse_pathbuf_info
}
```

Definition at line 872 of file ng_parse.c.

7.129.4.28 struct [ng_parse_type](#) [ng_parse_sizedstring_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_sizedstring_parse,
    ng_sizedstring_unparse,
    ng_sizedstring_getDefault,
    NULL
}
```

Definition at line 951 of file `ng_parse.c`.

7.129.4.29 struct [ng_parse_type](#) [ng_parse_string_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_string_parse,
    ng_string_unparse,
    ng_string_getDefault,
    NULL
}
```

Definition at line 779 of file `ng_parse.c`.

7.129.4.30 struct [ng_parse_type](#) [ng_parse_struct_type](#)**Initial value:**

```
{
    NULL,
    NULL,
    NULL,
    ng_struct_parse,
    ng_struct_unparse,
    ng_struct_getDefault,
    ng_struct_getAlign
}
```

Definition at line 223 of file `ng_parse.c`.

7.129.4.31 struct [ng_parse_fixedstring_info](#) [ng_parse_typebuf_info](#)**Initial value:**

```
{
    NG_TYPESIZ
}
```

Definition at line 877 of file `ng_parse.c`.

7.129.4.32 struct [ng_parse_type](#) [ng_parse_typebuf_type](#)**Initial value:**

```
{
    &ng_parse_fixedstring_type,
    &ng_parse_typebuf_info
}
```

Definition at line 880 of file `ng_parse.c`.

7.129.4.33 struct [ng_parse_type](#) [ng_parse_uint16_type](#)**Initial value:**

```
{
    &ng_parse_int16_type,
    (void *)INT_UNSIGNED
}
```

Definition at line 516 of file `ng_parse.c`.

7.129.4.34 struct [ng_parse_type](#) [ng_parse_uint32_type](#)**Initial value:**

```
{
    &ng_parse_int32_type,
    (void *)INT_UNSIGNED
}
```

Definition at line 618 of file `ng_parse.c`.

7.129.4.35 struct [ng_parse_type](#) [ng_parse_uint64_type](#)**Initial value:**

```
{
    &ng_parse_int64_type,
    (void *)INT_UNSIGNED
}
```

Definition at line 716 of file `ng_parse.c`.

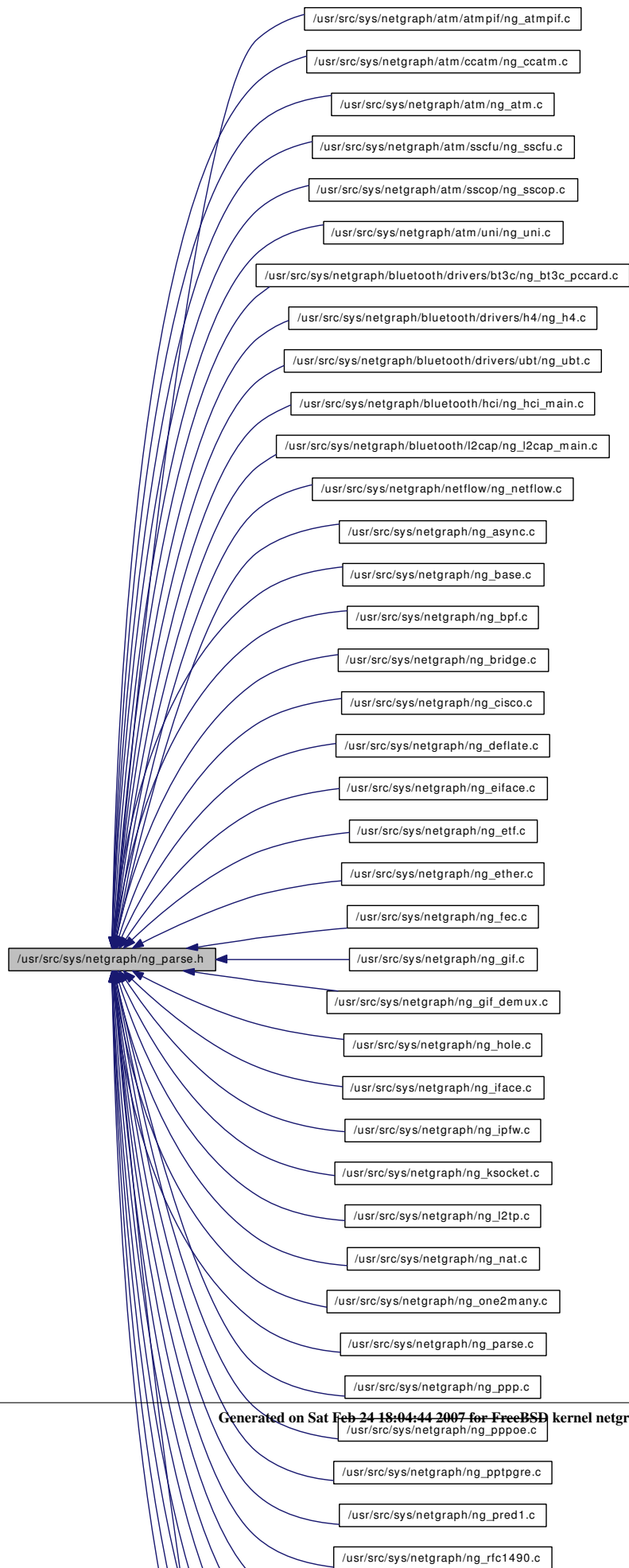
7.129.4.36 struct [ng_parse_type](#) [ng_parse_uint8_type](#)**Initial value:**

```
{
    &ng_parse_int8_type,
    (void *)INT_UNSIGNED
}
```

Definition at line 417 of file `ng_parse.c`.

7.130 /usr/src/sys/netgraph/ng_parse.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_parse_type](#)
- struct [ng_parse_struct_field](#)
- struct [ng_parse_fixedarray_info](#)
- struct [ng_parse_array_info](#)
- struct [ng_parse_fixedstring_info](#)

Typedefs

- typedef int [ng_parse_t](#) (const struct [ng_parse_type](#) *type, const char *string, int *off, const u_char *start, u_char *buf, int *buflen)
- typedef int [ng_unparse_t](#) (const struct [ng_parse_type](#) *type, const u_char *data, int *off, char *buf, int buflen)
- typedef int [ng_getDefault_t](#) (const struct [ng_parse_type](#) *type, const u_char *start, u_char *buf, int *buflen)
- typedef int [ng_getAlign_t](#) (const struct [ng_parse_type](#) *type)
- typedef int [ng_parse_array_getDefault_t](#) (const struct [ng_parse_type](#) *type, int index, const u_char *start, u_char *buf, int *buflen)
- typedef int [ng_parse_array_getLength_t](#) (const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)

Enumerations

- enum [ng_parse_token](#) {
 [T_LBRACE](#), [T_RBRACE](#), [T_LBRACKET](#), [T_RBRACKET](#),
 [T_EQUALS](#), [T_STRING](#), [T_ERROR](#), [T_WORD](#),
 [T_EOF](#) }

Functions

- int [ng_parse](#) (const struct [ng_parse_type](#) *type, const char *string, int *off, u_char *buf, int *buflen)
- int [ng_unparse](#) (const struct [ng_parse_type](#) *type, const u_char *data, char *buf, int buflen)
- int [ng_parse_getDefault](#) (const struct [ng_parse_type](#) *type, u_char *buf, int *buflen)
- enum [ng_parse_token](#) [ng_parse_get_token](#) (const char *s, int *startp, int *lenp)
- char * [ng_get_string_token](#) (const char *s, int *startp, int *lenp, int *slenp)
- char * [ng_encode_string](#) (const char *s, int slen)

Variables

- [ng_parse_type](#) [ng_parse_struct_type](#)
- [ng_parse_type](#) [ng_parse_fixedarray_type](#)
- [ng_parse_type](#) [ng_parse_array_type](#)
- [ng_parse_type](#) [ng_parse_string_type](#)
- [ng_parse_type](#) [ng_parse_fixedstring_type](#)
- [ng_parse_type](#) [ng_parse_sizedstring_type](#)
- [ng_parse_type](#) [ng_parse_nodebuf_type](#)
- [ng_parse_type](#) [ng_parse_hookbuf_type](#)

- [ng_parse_type ng_parse_pathbuf_type](#)
- [ng_parse_type ng_parse_typebuf_type](#)
- [ng_parse_type ng_parse_cmdbuf_type](#)
- [ng_parse_type ng_parse_int8_type](#)
- [ng_parse_type ng_parse_int16_type](#)
- [ng_parse_type ng_parse_int32_type](#)
- [ng_parse_type ng_parse_int64_type](#)
- [ng_parse_type ng_parse_uint8_type](#)
- [ng_parse_type ng_parse_uint16_type](#)
- [ng_parse_type ng_parse_uint32_type](#)
- [ng_parse_type ng_parse_uint64_type](#)
- [ng_parse_type ng_parse_hint8_type](#)
- [ng_parse_type ng_parse_hint16_type](#)
- [ng_parse_type ng_parse_hint32_type](#)
- [ng_parse_type ng_parse_hint64_type](#)
- [ng_parse_type ng_parse_ipaddr_type](#)
- [ng_parse_type ng_parse_enaddr_type](#)
- [ng_parse_type ng_parse_bytearray_type](#)
- [ng_parse_type ng_parse_ng_mesg_type](#)

7.130.1 Typedef Documentation

7.130.1.1 typedef int [ng_getAlign_t](#)(const struct [ng_parse_type](#) *type)

Definition at line 261 of file [ng_parse.h](#).

7.130.1.2 typedef int [ng_getDefault_t](#)(const struct [ng_parse_type](#) *type, const u_char *start, u_char *buf, int *buflen)

Definition at line 255 of file [ng_parse.h](#).

7.130.1.3 typedef int [ng_parse_array_getDefault_t](#)(const struct [ng_parse_type](#) *type, int index, const u_char *start, u_char *buf, int *buflen)

Definition at line 333 of file [ng_parse.h](#).

7.130.1.4 typedef int [ng_parse_array_getLength_t](#)(const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)

Definition at line 360 of file [ng_parse.h](#).

7.130.1.5 typedef int [ng_parse_t](#)(const struct [ng_parse_type](#) *type, const char *string, int *off, const u_char *start, u_char *buf, int *buflen)

Definition at line 226 of file [ng_parse.h](#).

7.130.1.6 typedef int **ng_unparse_t**(const struct **ng_parse_type** *type, const u_char *data, int *off, char *buf, int buflen)

Definition at line 241 of file ng_parse.h.

7.130.2 Enumeration Type Documentation

7.130.2.1 enum **ng_parse_token**

Enumerator:

T_LBRACE
T_RBRACE
T_LBRACKET
T_RBRACKET
T_EQUALS
T_STRING
T_ERROR
T_WORD
T_EOF

Definition at line 480 of file ng_parse.h.

7.130.3 Function Documentation

7.130.3.1 char* **ng_encode_string** (const char * s, int slen)

Definition at line 1825 of file ng_parse.c.

References M_NETGRAPH_PARSE.

Referenced by ng_ksocket_sockaddr_unparse(), ng_sizedstring_unparse(), and ng_string_unparse().

7.130.3.2 char* **ng_get_string_token** (const char * s, int * startp, int * lenp, int * slenp)

Definition at line 1737 of file ng_parse.c.

References M_NETGRAPH_PARSE.

Referenced by ng_bytearray_parse(), ng_fixedstring_parse(), ng_ksocket_sockaddr_parse(), ng_parse_get_token(), ng_sizedstring_parse(), and ng_string_parse().

7.130.3.3 int **ng_parse** (const struct **ng_parse_type** * type, const char * string, int * off, u_char * buf, int * buflen)

Definition at line 147 of file ng_parse.c.

References INVOKE, and ng_parse_type::parse.

Referenced by ng_generic_msg().

7.130.3.4 enum [ng_parse_token](#) [ng_parse_get_token](#) (const char * s, int * startp, int * lenp)

Definition at line 1691 of file ng_parse.c.

References M_NETGRAPH_PARSE, [ng_get_string_token\(\)](#), T_EOF, T_EQUALS, T_ERROR, T_LBRACE, T_LBRACKET, T_RBRACE, T_RBRACKET, T_STRING, and T_WORD.

Referenced by [ng_ksocket_sockaddr_parse\(\)](#), [ng_parse_composite\(\)](#), and [ng_parse_skip_value\(\)](#).

Here is the call graph for this function:



7.130.3.5 int [ng_parse_getDefault](#) (const struct [ng_parse_type](#) * type, u_char * buf, int * buflen)

Definition at line 169 of file ng_parse.c.

References [ng_parse_type::getDefault](#), and METHOD.

7.130.3.6 int [ng_unparse](#) (const struct [ng_parse_type](#) * type, const u_char * data, char * buf, int buflen)

Definition at line 157 of file ng_parse.c.

References INVOKE, and [ng_parse_type::unparse](#).

Referenced by [ng_generic_msg\(\)](#).

7.130.4 Variable Documentation

7.130.4.1 struct [ng_parse_type](#) [ng_parse_array_type](#)

Definition at line 319 of file ng_parse.c.

7.130.4.2 struct [ng_parse_type](#) [ng_parse_bytearray_type](#)

Definition at line 1164 of file ng_parse.c.

7.130.4.3 struct [ng_parse_type](#) [ng_parse_cmdbuf_type](#)

Definition at line 888 of file ng_parse.c.

7.130.4.4 struct [ng_parse_type](#) [ng_parse_enaddr_type](#)

Definition at line 1069 of file ng_parse.c.

7.130.4.5 struct [ng_parse_type](#) [ng_parse_fixedarray_type](#)

Definition at line 272 of file ng_parse.c.

7.130.4.6 struct [ng_parse_type](#) [ng_parse_fixedstring_type](#)

Definition at line 843 of file `ng_parse.c`.

7.130.4.7 struct [ng_parse_type](#) [ng_parse_hint16_type](#)

Definition at line 521 of file `ng_parse.c`.

7.130.4.8 struct [ng_parse_type](#) [ng_parse_hint32_type](#)

Definition at line 623 of file `ng_parse.c`.

7.130.4.9 struct [ng_parse_type](#) [ng_parse_hint64_type](#)

Definition at line 721 of file `ng_parse.c`.

7.130.4.10 struct [ng_parse_type](#) [ng_parse_hint8_type](#)

Definition at line 422 of file `ng_parse.c`.

7.130.4.11 struct [ng_parse_type](#) [ng_parse_hookbuf_type](#)

Definition at line 864 of file `ng_parse.c`.

7.130.4.12 struct [ng_parse_type](#) [ng_parse_int16_type](#)

Definition at line 506 of file `ng_parse.c`.

7.130.4.13 struct [ng_parse_type](#) [ng_parse_int32_type](#)

Definition at line 608 of file `ng_parse.c`.

7.130.4.14 struct [ng_parse_type](#) [ng_parse_int64_type](#)

Definition at line 706 of file `ng_parse.c`.

7.130.4.15 struct [ng_parse_type](#) [ng_parse_int8_type](#)

Definition at line 407 of file `ng_parse.c`.

Referenced by `ng_ipaddr_parse()`.

7.130.4.16 struct [ng_parse_type](#) [ng_parse_ipaddr_type](#)

Definition at line 1013 of file `ng_parse.c`.

7.130.4.17 struct [ng_parse_type](#) [ng_parse_ng_mesg_type](#)

Definition at line 1198 of file `ng_parse.c`.

7.130.4.18 struct [ng_parse_type](#) [ng_parse_nodebuf_type](#)

Definition at line 856 of file `ng_parse.c`.

7.130.4.19 struct [ng_parse_type](#) [ng_parse_pathbuf_type](#)

Definition at line 872 of file `ng_parse.c`.

7.130.4.20 struct [ng_parse_type](#) [ng_parse_sizedstring_type](#)

Definition at line 951 of file `ng_parse.c`.

7.130.4.21 struct [ng_parse_type](#) [ng_parse_string_type](#)

Definition at line 779 of file `ng_parse.c`.

7.130.4.22 struct [ng_parse_type](#) [ng_parse_struct_type](#)

Definition at line 223 of file `ng_parse.c`.

7.130.4.23 struct [ng_parse_type](#) [ng_parse_typebuf_type](#)

Definition at line 880 of file `ng_parse.c`.

7.130.4.24 struct [ng_parse_type](#) [ng_parse_uint16_type](#)

Definition at line 516 of file `ng_parse.c`.

7.130.4.25 struct [ng_parse_type](#) [ng_parse_uint32_type](#)

Definition at line 618 of file `ng_parse.c`.

7.130.4.26 struct [ng_parse_type](#) [ng_parse_uint64_type](#)

Definition at line 716 of file `ng_parse.c`.

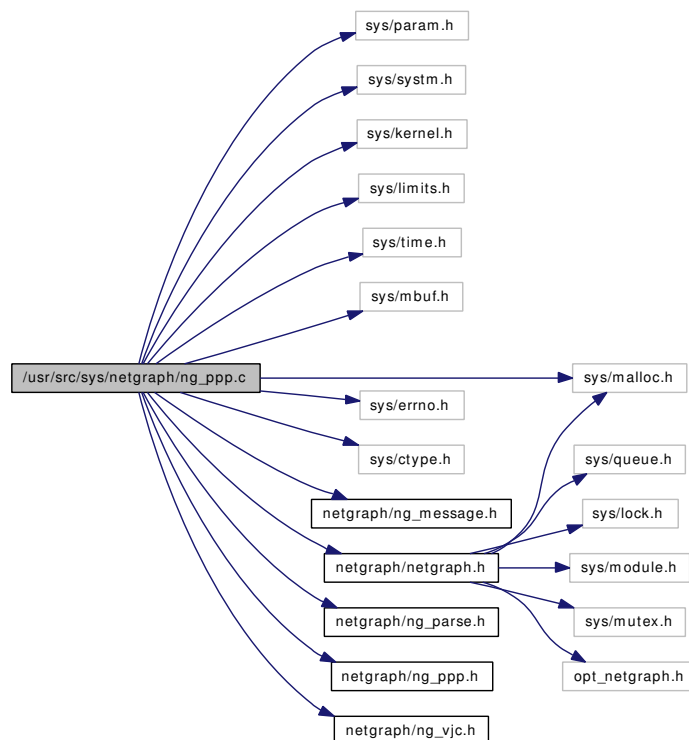
7.130.4.27 struct [ng_parse_type](#) [ng_parse_uint8_type](#)

Definition at line 417 of file `ng_parse.c`.

7.131 /usr/src/sys/netgraph/ng_ppp.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/limits.h>
#include <sys/time.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/ctype.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_ppp.h>
#include <netgraph/ng_vjc.h>
```

Include dependency graph for ng_ppp.c:



Data Structures

- struct [ng_ppp_frag](#)

- struct [ng_ppp_link](#)
- struct [ng_ppp_private](#)

Defines

- #define [M_NETGRAPH_PPP](#) M_NETGRAPH
- #define [PROT_VALID](#)(p) (((p) & 0x0101) == 0x0001)
- #define [PROT_COMPRESSABLE](#)(p) (((p) & 0xff00) == 0x0000)
- #define [PROT_ATALK](#) 0x0029
- #define [PROT_COMPD](#) 0x00fd
- #define [PROT_CRYPTD](#) 0x0053
- #define [PROT_IP](#) 0x0021
- #define [PROT_IPV6](#) 0x0057
- #define [PROT_IPX](#) 0x002b
- #define [PROT_LCP](#) 0xc021
- #define [PROT_MP](#) 0x003d
- #define [PROT_VJCOMP](#) 0x002d
- #define [PROT_VJUNCOMP](#) 0x002f
- #define [MP_MIN_MRRU](#) 1500
- #define [MP_INITIAL_SEQ](#) 0
- #define [MP_MIN_LINK_MRU](#) 32
- #define [MP_SHORT_SEQ_MASK](#) 0x00000fff
- #define [MP_SHORT_SEQ_HIBIT](#) 0x00000800
- #define [MP_SHORT_FIRST_FLAG](#) 0x00008000
- #define [MP_SHORT_LAST_FLAG](#) 0x00004000
- #define [MP_LONG_SEQ_MASK](#) 0x00ffffff
- #define [MP_LONG_SEQ_HIBIT](#) 0x00800000
- #define [MP_LONG_FIRST_FLAG](#) 0x80000000
- #define [MP_LONG_LAST_FLAG](#) 0x40000000
- #define [MP_NOSEQ](#) 0x7fffffff
- #define [MP_SHORT_EXTEND](#)(s)
- #define [MP_LONG_EXTEND](#)(s)
- #define [MP_SHORT_SEQ_DIFF](#)(x, y) MP_SHORT_EXTEND((x) - (y))
- #define [MP_LONG_SEQ_DIFF](#)(x, y) MP_LONG_EXTEND((x) - (y))
- #define [MP_RECV_SEQ_DIFF](#)(priv, x, y)
- #define [MP_NEXT_RECV_SEQ](#)(priv, seq)
- #define [MP_MIN_FRAG_LEN](#) 32
- #define [MP_MAX_QUEUE_LEN](#) 128
- #define [MP_FRAGTIMER_INTERVAL](#) (hz/2)
- #define [HOOK_INDEX_MAX](#) 13
- #define [HOOK_INDEX_ATALK](#) 0
- #define [HOOK_INDEX_BYPASS](#) 1
- #define [HOOK_INDEX_COMPRESS](#) 2
- #define [HOOK_INDEX_ENCRYPT](#) 3
- #define [HOOK_INDEX_DECOMPRESS](#) 4
- #define [HOOK_INDEX_DECRYPT](#) 5
- #define [HOOK_INDEX_INET](#) 6
- #define [HOOK_INDEX_IPX](#) 7
- #define [HOOK_INDEX_VJC_COMP](#) 8
- #define [HOOK_INDEX_VJC_IP](#) 9

- #define [HOOK_INDEX_VJC_UNCOMP](#) 10
- #define [HOOK_INDEX_VJC_VJIP](#) 11
- #define [HOOK_INDEX_IPV6](#) 12
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef [ng_ppp_private](#) * [priv_p](#)

Functions

- static int [ng_ppp_proto_rcv](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_hcomp_xmit](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto)
- static int [ng_ppp_hcomp_rcv](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_comp_xmit](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto)
- static int [ng_ppp_comp_rcv](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_crypt_xmit](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto)
- static int [ng_ppp_crypt_rcv](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_mp_xmit](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto)
- static int [ng_ppp_mp_rcv](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_link_xmit](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_bypass](#) ([node_p](#) node, [item_p](#) item, [uint16_t](#) proto, [uint16_t](#) linkNum)
- static int [ng_ppp_check_packet](#) ([node_p](#) node)
- static void [ng_ppp_get_packet](#) ([node_p](#) node, struct mbuf **mp)
- static int [ng_ppp_frag_process](#) ([node_p](#) node)
- static int [ng_ppp_frag_trim](#) ([node_p](#) node)
- static void [ng_ppp_frag_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static void [ng_ppp_frag_checkstale](#) ([node_p](#) node)
- static void [ng_ppp_frag_reset](#) ([node_p](#) node)
- static void [ng_ppp_mp_strategy](#) ([node_p](#) node, int len, int *distrib)
- static int [ng_ppp_intcmp](#) (void *latency, const void *v1, const void *v2)
- static struct mbuf * [ng_ppp_addproto](#) (struct mbuf *m, [uint16_t](#) proto, int compOK)
- static struct mbuf * [ng_ppp_cutproto](#) (struct mbuf *m, [uint16_t](#) *proto)
- static struct mbuf * [ng_ppp_prepend](#) (struct mbuf *m, const void *buf, int len)
- static int [ng_ppp_config_valid](#) ([node_p](#) node, const struct [ng_ppp_node_conf](#) *newConf)
- static void [ng_ppp_update](#) ([node_p](#) node, int newConf)
- static void [ng_ppp_start_frag_timer](#) ([node_p](#) node)
- static void [ng_ppp_stop_frag_timer](#) ([node_p](#) node)
- [NETGRAPH_INIT](#) (ppp,&[ng_ppp_typestruct](#))
- static int [ng_ppp_constructor](#) ([node_p](#) node)
- static int [ng_ppp_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_ppp_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_ppp_shutdown](#) ([node_p](#) node)
- static int [ng_ppp_disconnect](#) ([hook_p](#) hook)
- static int [ng_ppp_rcvdata_inet](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_ppp_rcvdata_ipv6](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_ppp_rcvdata_atalk](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_ppp_rcvdata_ipx](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_ppp_rcvdata_bypass](#) ([hook_p](#) hook, [item_p](#) item)

- static int `ng_ppp_rcvdata_vjc_comp` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_vjc_uncomp` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_vjc_vjip` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_vjc_ip` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_compress` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_decompress` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_encrypt` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata_decrypt` (`hook_p` hook, `item_p` item)
- static int `ng_ppp_rcvdata` (`hook_p` hook, `item_p` item)

Variables

- static `ng_constructor_t` `ng_ppp_constructor`
- static `ng_rcvmsg_t` `ng_ppp_rcvmsg`
- static `ng_shutdown_t` `ng_ppp_shutdown`
- static `ng_newhook_t` `ng_ppp_newhook`
- static `ng_rcvdata_t` `ng_ppp_rcvdata`
- static `ng_disconnect_t` `ng_ppp_disconnect`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_inet`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_ipv6`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_ipx`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_atalk`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_bypass`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_vjc_ip`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_vjc_comp`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_vjc_uncomp`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_vjc_vjip`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_compress`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_decompress`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_encrypt`
- static `ng_rcvdata_t` `ng_ppp_rcvdata_decrypt`
- struct {
 - char *const name
 - `ng_rcvdata_t` * fn
} `ng_ppp_hook_names` []
- static struct `ng_parse_fixedarray_info` `ng_ppp_rseq_array_info`
- static struct `ng_parse_type` `ng_ppp_rseq_array_type`
- static struct `ng_parse_struct_field` `ng_ppp_mp_state_type_fields` [] = `NG_PPP_MP_STATE_TYPE_INFO(&ng_ppp_rseq_array_type)`
- static struct `ng_parse_type` `ng_ppp_mp_state_type`
- static struct `ng_parse_struct_field` `ng_ppp_link_type_fields` [] = `NG_PPP_LINK_TYPE_INFO`
- static struct `ng_parse_type` `ng_ppp_link_type`
- static struct `ng_parse_struct_field` `ng_ppp_bund_type_fields` [] = `NG_PPP_BUND_TYPE_INFO`
- static struct `ng_parse_type` `ng_ppp_bund_type`
- static struct `ng_parse_fixedarray_info` `ng_ppp_array_info`
- static struct `ng_parse_type` `ng_ppp_link_array_type`
- static struct `ng_parse_struct_field` `ng_ppp_conf_type_fields` [] = `NG_PPP_CONFIG_TYPE_INFO(&ng_ppp_bund_type, &ng_ppp_link_array_type)`
- static struct `ng_parse_type` `ng_ppp_conf_type`

- static struct `ng_parse_struct_field ng_ppp_stats_type_fields [] = NG_PPP_STATS_TYPE_INFO`
- static struct `ng_parse_type ng_ppp_stats_type`
- static struct `ng_cmdlist ng_ppp_cmds []`
- static struct `ng_type ng_ppp_tpestruct`
- static const uint8_t `ng_ppp_acf [2] = { 0xff, 0x03 }`
- static struct timeval `ng_ppp_max_staleness = { 2, 0 }`

7.131.1 Define Documentation

7.131.1.1 `#define ERROUT(x) do { error = (x); goto done; } while (0)`

Definition at line 446 of file `ng_ppp.c`.

7.131.1.2 `#define HOOK_INDEX_ATALK 0`

Referenced by `ng_ppp_proto_recv()`.

7.131.1.3 `#define HOOK_INDEX_BYPASS 1`

Referenced by `ng_ppp_bypass()`.

7.131.1.4 `#define HOOK_INDEX_COMPRESS 2`

Referenced by `ng_ppp_comp_xmit()`.

7.131.1.5 `#define HOOK_INDEX_DECOMPRESS 4`

Referenced by `ng_ppp_comp_recv()`.

7.131.1.6 `#define HOOK_INDEX_DECRYPT 5`

Referenced by `ng_ppp_crypt_recv()`.

7.131.1.7 `#define HOOK_INDEX_ENCRYPT 3`

Referenced by `ng_ppp_crypt_xmit()`.

7.131.1.8 `#define HOOK_INDEX_INET 6`

Referenced by `ng_ppp_proto_recv()`.

7.131.1.9 `#define HOOK_INDEX_IPV6 12`

Referenced by `ng_ppp_proto_recv()`.

7.131.1.10 #define HOOK_INDEX_IPX 7

Referenced by ng_ppp_proto_rcv().

7.131.1.11 #define HOOK_INDEX_MAX 13

Definition at line 181 of file ng_ppp.c.

7.131.1.12 #define HOOK_INDEX_VJC_COMP 8

Referenced by ng_ppp_hcomp_rcv(), and ng_ppp_update().

7.131.1.13 #define HOOK_INDEX_VJC_IP 9

Referenced by ng_ppp_hcomp_xmit(), ng_ppp_rcvmsg(), and ng_ppp_update().

7.131.1.14 #define HOOK_INDEX_VJC_UNCOMP 10

Referenced by ng_ppp_hcomp_rcv(), and ng_ppp_update().

7.131.1.15 #define HOOK_INDEX_VJC_VJIP 11

Referenced by ng_ppp_update().

7.131.1.16 #define M_NETGRAPH_PPP M_NETGRAPH

Definition at line 112 of file ng_ppp.c.

Referenced by ng_ppp_constructor(), ng_ppp_frag_checkstale(), ng_ppp_frag_process(), ng_ppp_frag_reset(), ng_ppp_frag_trim(), ng_ppp_get_packet(), ng_ppp_mp_rcv(), and ng_ppp_shutdown().

7.131.1.17 #define MP_FRAGTIMER_INTERVAL (hz/2)

Definition at line 178 of file ng_ppp.c.

Referenced by ng_ppp_start_frag_timer().

7.131.1.18 #define MP_INITIAL_SEQ 0

Definition at line 132 of file ng_ppp.c.

Referenced by ng_ppp_update().

7.131.1.19 #define MP_LONG_EXTEND(s)**Value:**

```
((s) & MP_LONG_SEQ_HIBIT) ? \
    ((s) | ~MP_LONG_SEQ_MASK) \
    : ((s) & MP_LONG_SEQ_MASK)
```

Definition at line 151 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv().

7.131.1.20 #define MP_LONG_FIRST_FLAG 0x80000000

Definition at line 142 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv(), and ng_ppp_mp_xmit().

7.131.1.21 #define MP_LONG_LAST_FLAG 0x40000000

Definition at line 143 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv(), and ng_ppp_mp_xmit().

7.131.1.22 #define MP_LONG_SEQ_DIFF(x, y) MP_LONG_EXTEND((x) - (y))

Definition at line 158 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv().

7.131.1.23 #define MP_LONG_SEQ_HIBIT 0x00800000

Definition at line 141 of file ng_ppp.c.

7.131.1.24 #define MP_LONG_SEQ_MASK 0x00ffffff

Definition at line 140 of file ng_ppp.c.

Referenced by ng_ppp_mp_xmit().

7.131.1.25 #define MP_MAX_QUEUE_LEN 128

Definition at line 175 of file ng_ppp.c.

Referenced by ng_ppp_frag_process().

7.131.1.26 #define MP_MIN_FRAG_LEN 32

Definition at line 172 of file ng_ppp.c.

Referenced by ng_ppp_mp_xmit().

7.131.1.27 #define MP_MIN_LINK_MRU 32

Definition at line 133 of file ng_ppp.c.

Referenced by ng_ppp_config_valid().

7.131.1.28 #define MP_MIN_MRRU 1500

Definition at line 131 of file ng_ppp.c.

Referenced by ng_ppp_config_valid().

7.131.1.29 #define MP_NEXT_RECV_SEQ(priv, seq)**Value:**

```
((priv)->conf.recvShortSeq ?
      \
      MP_SHORT_EXTEND((seq) + 1) :      \
      MP_LONG_EXTEND((seq) + 1))
```

Definition at line 166 of file ng_ppp.c.

Referenced by ng_ppp_check_packet(), ng_ppp_frag_checkstale(), and ng_ppp_frag_trim().

7.131.1.30 #define MP_NOSEQ 0x7fffffff

Definition at line 145 of file ng_ppp.c.

Referenced by ng_ppp_constructor(), ng_ppp_rcvmsg(), and ng_ppp_update().

7.131.1.31 #define MP_RECV_SEQ_DIFF(priv, x, y)**Value:**

```
((priv)->conf.recvShortSeq ?
      \
      MP_SHORT_SEQ_DIFF((x), (y)) :      \
      MP_LONG_SEQ_DIFF((x), (y)))
```

Definition at line 160 of file ng_ppp.c.

Referenced by ng_ppp_check_packet(), ng_ppp_frag_checkstale(), ng_ppp_frag_process(), ng_ppp_frag_trim(), and ng_ppp_mp_recv().

7.131.1.32 #define MP_SHORT_EXTEND(s)**Value:**

```
((s) & MP_SHORT_SEQ_HIBIT) ?
      \
      ((s) | ~MP_SHORT_SEQ_MASK) :      \
      ((s) & MP_SHORT_SEQ_MASK)
```

Definition at line 148 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv().

7.131.1.33 #define MP_SHORT_FIRST_FLAG 0x00008000

Definition at line 137 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv(), and ng_ppp_mp_xmit().

7.131.1.34 #define MP_SHORT_LAST_FLAG 0x00004000

Definition at line 138 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv(), and ng_ppp_mp_xmit().

7.131.1.35 #define MP_SHORT_SEQ_DIFF(x, y) MP_SHORT_EXTEND((x) - (y))

Definition at line 157 of file ng_ppp.c.

Referenced by ng_ppp_mp_recv().

7.131.1.36 #define MP_SHORT_SEQ_HIBIT 0x00000800

Definition at line 136 of file ng_ppp.c.

7.131.1.37 #define MP_SHORT_SEQ_MASK 0x00000fff

Definition at line 135 of file ng_ppp.c.

Referenced by ng_ppp_mp_xmit().

7.131.1.38 #define PROT_ATALK 0x0029

Definition at line 119 of file ng_ppp.c.

Referenced by ng_ppp_proto_recv(), and ng_ppp_rcvdata_atalk().

7.131.1.39 #define PROT_COMPD 0x00fd

Definition at line 120 of file ng_ppp.c.

7.131.1.40 #define PROT_COMPRESSABLE(p) (((p) & 0xff00) == 0x0000)

Definition at line 116 of file ng_ppp.c.

Referenced by ng_ppp_addproto().

7.131.1.41 #define PROT_CRYPTD 0x0053

Definition at line 121 of file ng_ppp.c.

Referenced by ng_ppp_comp_xmit(), ng_ppp_crypt_recv(), ng_ppp_crypt_xmit(), and ng_ppp_rcvdata_decrypt().

7.131.1.42 #define PROT_IP 0x0021

Definition at line 122 of file ng_ppp.c.

Referenced by ng_ppp_hcomp_xmit(), ng_ppp_proto_recv(), ng_ppp_rcvdata_inet(), ng_ppp_rcvdata_vjc_ip(), and ng_ppp_rcvdata_vjc_vjip().

7.131.1.43 #define PROT_IPV6 0x0057

Definition at line 123 of file ng_ppp.c.

Referenced by ng_ppp_proto_rcv(), and ng_ppp_rcvdata_ipv6().

7.131.1.44 #define PROT_IPX 0x002b

Definition at line 124 of file ng_ppp.c.

Referenced by ng_ppp_proto_rcv(), and ng_ppp_rcvdata_ipx().

7.131.1.45 #define PROT_LCP 0xc021

Definition at line 125 of file ng_ppp.c.

Referenced by ng_ppp_link_xmit().

7.131.1.46 #define PROT_MP 0x003d

Definition at line 126 of file ng_ppp.c.

Referenced by ng_ppp_mp_rcv(), and ng_ppp_mp_xmit().

7.131.1.47 #define PROT_VALID(p) (((p) & 0x0101) == 0x0001)

Definition at line 115 of file ng_ppp.c.

Referenced by ng_ppp_cutproto(), ng_ppp_frag_checkstale(), ng_ppp_frag_process(), ng_ppp_rcvdata(), ng_ppp_rcvdata_compress(), ng_ppp_rcvdata_decompress(), and ng_ppp_rcvdata_decrypt().

7.131.1.48 #define PROT_VJCOMP 0x002d

Definition at line 127 of file ng_ppp.c.

Referenced by ng_ppp_hcomp_rcv(), and ng_ppp_rcvdata_vjc_comp().

7.131.1.49 #define PROT_VJUNCOMP 0x002f

Definition at line 128 of file ng_ppp.c.

Referenced by ng_ppp_hcomp_rcv(), and ng_ppp_rcvdata_vjc_uncomp().

7.131.2 Typedef Documentation**7.131.2.1 typedef struct [ng_ppp_private](#)* [priv_p](#)**

Definition at line 222 of file ng_ppp.c.

7.131.3 Function Documentation

7.131.3.1 NETGRAPH_INIT (ppp, & ng_ppp_tpestruct)

7.131.3.2 static struct mbuf * ng_ppp_addproto (struct mbuf * m, uint16_t proto, int compOK) [static]

Definition at line 2222 of file ng_ppp.c.

References ng_ppp_prepend(), and PROT_COMPRESSABLE.

Referenced by ng_ppp_comp_recv(), ng_ppp_comp_xmit(), ng_ppp_crypt_xmit(), ng_ppp_link_xmit(), and ng_ppp_mp_xmit().

Here is the call graph for this function:



7.131.3.3 static int ng_ppp_bypass (node_p node, item_p item, uint16_t proto, uint16_t linkNum) [static]

Definition at line 814 of file ng_ppp.c.

References HOOK_INDEX_BYPASS, NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, ng_ppp_prepend(), NGL_GET_M, and NGL_M.

Referenced by ng_ppp_comp_recv(), ng_ppp_crypt_recv(), ng_ppp_proto_recv(), and ng_ppp_rcvdata().

Here is the call graph for this function:



7.131.3.4 static int ng_ppp_check_packet (node_p node) [static]

Definition at line 1526 of file ng_ppp.c.

References ng_ppp_frag::first, ng_ppp_frag::last, MP_NEXT_RECV_SEQ, MP_RECV_SEQ_DIFF, NG_NODE_PRIVATE, and ng_ppp_frag::seq.

Referenced by ng_ppp_frag_process().

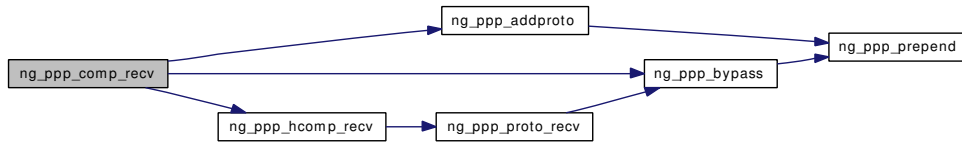
7.131.3.5 static int ng_ppp_comp_recv (node_p node, item_p item, uint16_t proto, uint16_t linkNum) [static]

Definition at line 1061 of file ng_ppp.c.

References HOOK_INDEX_DECOMPRESS, NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, ng_ppp_addproto(), ng_ppp_bypass(), NG_PPP_DECOMPRESS_FULL, ng_ppp_hcomp_recv(), NGL_GET_M, NGL_M, and PROT_COMPD.

Referenced by ng_ppp_crypt_recv(), and ng_ppp_rcvdata_decrypt().

Here is the call graph for this function:



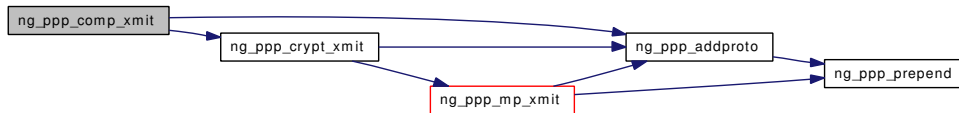
7.131.3.6 static int ng_ppp_comp_xmit (node_p node, item_p item, uint16_t proto) [static]

Definition at line 996 of file ng_ppp.c.

References HOOK_INDEX_COMPRESS, NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, ng_ppp_addproto(), ng_ppp_crypt_xmit(), NGI_GET_M, NGI_M, PROT_COMPD, and PROT_CRYPTD.

Referenced by ng_ppp_hcomp_xmit(), ng_ppp_rcvdata_vjc_comp(), ng_ppp_rcvdata_vjc_uncomp(), and ng_ppp_rcvdata_vjc_vjip().

Here is the call graph for this function:



7.131.3.7 static int ng_ppp_config_valid (node_p node, const struct ng_ppp_node_conf * newConf) [static]

Definition at line 2355 of file ng_ppp.c.

References ng_ppp_link_conf::bandwidth, ng_ppp_node_conf::bund, ng_ppp_link_conf::enableLink, ng_ppp_bund_conf::enableMultilink, ng_ppp_link_conf::latency, ng_ppp_node_conf::links, MP_MIN_LINK_MRU, MP_MIN_MRRU, ng_ppp_bund_conf::mrru, ng_ppp_link_conf::mru, NG_NODE_PRIVATE, NG_PPP_MAX_BANDWIDTH, NG_PPP_MAX_LATENCY, NG_PPP_MAX_LINKS, ng_ppp_bund_conf::recvShortSeq, and ng_ppp_bund_conf::xmitShortSeq.

Referenced by ng_ppp_rcvmsg().

7.131.3.8 static int ng_ppp_constructor (node_p node) [static]

Definition at line 456 of file ng_ppp.c.

References M_NETGRAPH_PPP, MP_NOSEQ, ng_callout_init, NG_NODE_SET_PRIVATE, and NG_PPP_MAX_LINKS.

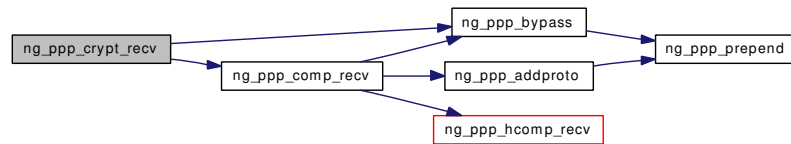
7.131.3.9 static int ng_ppp_crypt_rcv (node_p node, item_p item, uint16_t proto, uint16_t linkNum) [static]

Definition at line 1171 of file ng_ppp.c.

References HOOK_INDEX_DECRYPT, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, ng_ppp_bypass(), ng_ppp_comp_rcv(), NGI_M, and PROT_CRYPTD.

Referenced by `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, and `ng_ppp_mp_rcv()`.

Here is the call graph for this function:



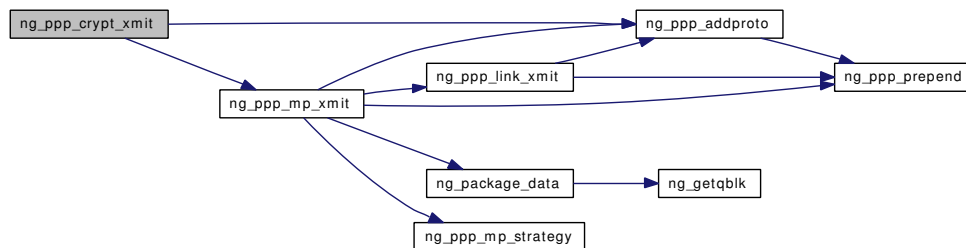
7.131.3.10 `static int ng_ppp_crypt_xmit (node_p node, item_p item, uint16_t proto)` [static]

Definition at line 1128 of file `ng_ppp.c`.

References `HOOK_INDEX_ENCRYPT`, `NG_FREE_ITEM`, `NG_FWD_ITEM_HOOK`, `NG_NODE_PRIVATE`, `ng_ppp_addproto()`, `ng_ppp_mp_xmit()`, `NGI_GET_M`, `NGI_M`, and `PROT_CRYPTD`.

Referenced by `ng_ppp_comp_xmit()`, and `ng_ppp_rcvdata_compress()`.

Here is the call graph for this function:



7.131.3.11 `static struct mbuf * ng_ppp_cutproto (struct mbuf * m, uint16_t * proto)` [static]

Definition at line 2239 of file `ng_ppp.c`.

References `PROT_VALID`.

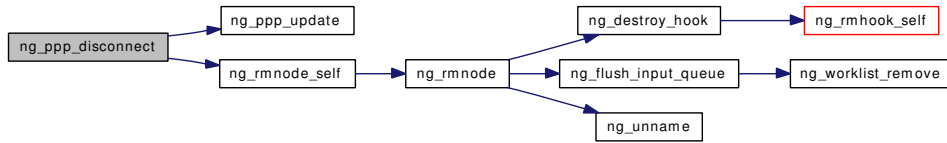
Referenced by `ng_ppp_frag_checkstale()`, `ng_ppp_frag_process()`, `ng_ppp_rcvdata()`, `ng_ppp_rcvdata_compress()`, `ng_ppp_rcvdata_decompress()`, and `ng_ppp_rcvdata_decrypt()`.

7.131.3.12 `static int ng_ppp_disconnect (hook_p hook)` [static]

Definition at line 693 of file `ng_ppp.c`.

References `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, `ng_ppp_update()`, and `ng_rmnode_self()`.

Here is the call graph for this function:



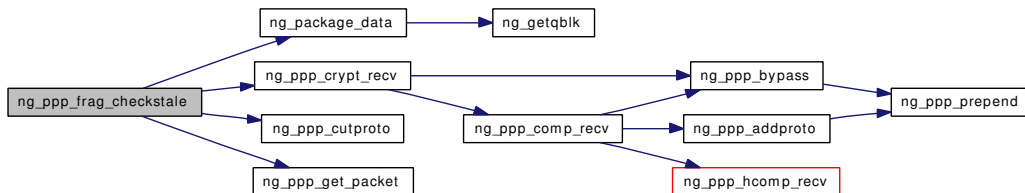
7.131.3.13 static void ng_ppp_frag_checkstale (node_p node) [static]

Definition at line 1734 of file ng_ppp.c.

References ng_ppp_frag::data, ng_ppp_frag::first, ng_ppp_frag::last, M_NETGRAPH_PPP, MP_NEXT_RECV_SEQ, MP_RECV_SEQ_DIFF, NG_FREE_M, NG_NODE_PRIVATE, NG_NOFLAGS, ng_package_data(), NG_PPP_BUNDLE_LINKNUM, ng_ppp_crypt_rcv(), ng_ppp_cutproto(), ng_ppp_get_packet(), PROT_VALID, ng_ppp_frag::seq, and ng_ppp_frag::timestamp.

Referenced by ng_ppp_frag_timeout().

Here is the call graph for this function:



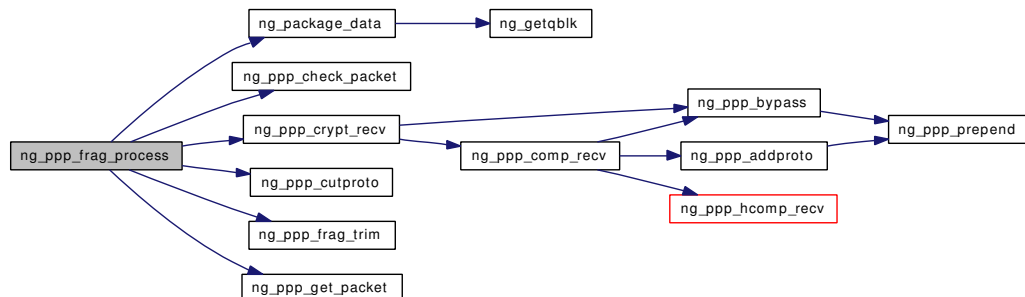
7.131.3.14 static int ng_ppp_frag_process (node_p node) [static]

Definition at line 1644 of file ng_ppp.c.

References ng_ppp_frag::data, M_NETGRAPH_PPP, MP_MAX_QUEUE_LEN, MP_RECV_SEQ_DIFF, NG_FREE_M, NG_NODE_PRIVATE, NG_NOFLAGS, ng_package_data(), NG_PPP_BUNDLE_LINKNUM, ng_ppp_check_packet(), ng_ppp_crypt_rcv(), ng_ppp_cutproto(), ng_ppp_frag_trim(), ng_ppp_get_packet(), PROT_VALID, and ng_ppp_frag::seq.

Referenced by ng_ppp_mp_rcv().

Here is the call graph for this function:



7.131.3.15 static void ng_ppp_frag_reset (node_p node) [static]

Definition at line 2401 of file ng_ppp.c.

References ng_ppp_frag::data, M_NETGRAPH_PPP, NG_FREE_M, and NG_NODE_PRIVATE.

Referenced by ng_ppp_shutdown().

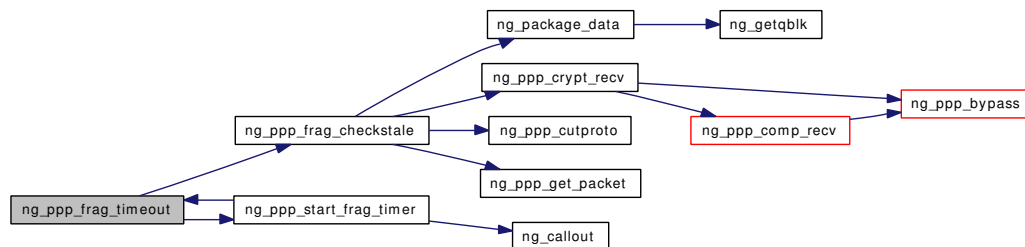
7.131.3.16 static void ng_ppp_frag_timeout (node_p node, hook_p hook, void * arg1, int arg2) [static]

Definition at line 1828 of file ng_ppp.c.

References NG_NODE_NOT_VALID, ng_ppp_frag_checkstale(), and ng_ppp_start_frag_timer().

Referenced by ng_ppp_start_frag_timer().

Here is the call graph for this function:

**7.131.3.17 static int ng_ppp_frag_trim (node_p node) [static]**

Definition at line 1595 of file ng_ppp.c.

References ng_ppp_frag::data, ng_ppp_frag::first, ng_ppp_frag::last, M_NETGRAPH_PPP, MP_NEXT_RECVC_SEQ, MP_RECV_SEQ_DIFF, NG_FREE_M, NG_NODE_PRIVATE, and ng_ppp_frag::seq.

Referenced by ng_ppp_frag_process().

7.131.3.18 static void ng_ppp_get_packet (node_p node, struct mbuf ** mp) [static]

Definition at line 1559 of file ng_ppp.c.

References ng_ppp_frag::data, ng_ppp_frag::first, ng_ppp_frag::last, M_NETGRAPH_PPP, and NG_NODE_PRIVATE.

Referenced by ng_ppp_frag_checkstale(), and ng_ppp_frag_process().

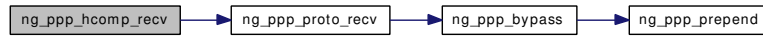
7.131.3.19 static int ng_ppp_hcomp_rcv (node_p node, item_p item, uint16_t proto, uint16_t linkNum) [static]

Definition at line 947 of file ng_ppp.c.

References HOOK_INDEX_VJC_COMP, HOOK_INDEX_VJC_UNCOMP, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, ng_ppp_proto_rcvc(), PROT_VJCOMP, and PROT_VJUNCOMP.

Referenced by ng_ppp_comp_rcvc(), and ng_ppp_rcvdata_decompress().

Here is the call graph for this function:



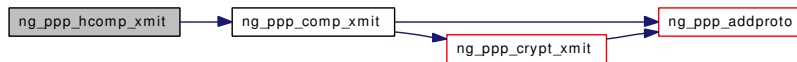
7.131.3.20 `static int ng_ppp_hcomp_xmit (node_p node, item_p item, uint16_t proto)` [static]

Definition at line 881 of file ng_ppp.c.

References `HOOK_INDEX_VJC_IP`, `NG_FWD_ITEM_HOOK`, `NG_NODE_PRIVATE`, `ng_ppp_comp_xmit()`, and `PROT_IP`.

Referenced by `ng_ppp_rcvdata_atalk()`, `ng_ppp_rcvdata_bypass()`, `ng_ppp_rcvdata_inet()`, `ng_ppp_rcvdata_ipv6()`, and `ng_ppp_rcvdata_ipx()`.

Here is the call graph for this function:



7.131.3.21 `static int ng_ppp_intcmp (void *latency, const void *v1, const void *v2)` [static]

Definition at line 2210 of file ng_ppp.c.

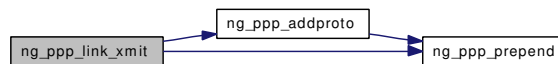
7.131.3.22 `static int ng_ppp_link_xmit (node_p node, item_p item, uint16_t proto, uint16_t linkNum)` [static]

Definition at line 1232 of file ng_ppp.c.

References `ng_ppp_link::bytesInQueue`, `ng_ppp_link::conf`, `ng_ppp_link_conf::enableACFComp`, `ng_ppp_link_conf::enableProtoComp`, `ng_ppp_link::hook`, `ng_ppp_link::lastWrite`, `ng_ppp_link_conf::mru`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_NODE_PRIVATE`, `ng_ppp_acf`, `ng_ppp_addproto()`, `NG_PPP_MAX_LINKS`, `ng_ppp_prepend()`, `NGI_GET_M`, `PROT_LCP`, `ng_ppp_link::stats`, `ng_ppp_link_stat::xmitFrames`, and `ng_ppp_link_stat::xmitOctets`.

Referenced by `ng_ppp_mp_xmit()`, and `ng_ppp_rcvdata_bypass()`.

Here is the call graph for this function:



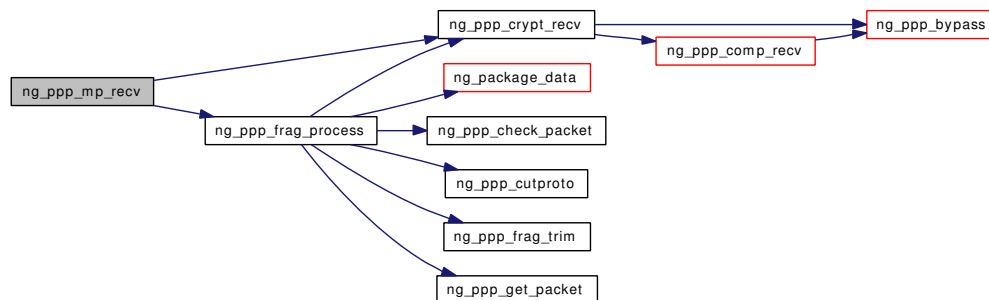
7.131.3.23 `static int ng_ppp_mp_recv (node_p node, item_p item, uint16_t proto, uint16_t linkNum)` [static]

Definition at line 1412 of file ng_ppp.c.

References `ng_ppp_frag::data`, `ng_ppp_link_stat::dropFragments`, `ng_ppp_link_stat::dupFragments`, `ng_ppp_frag::first`, `ng_ppp_frag::last`, `M_NETGRAPH_PPP`, `MP_LONG_EXTEND`, `MP_LONG_FIRST_FLAG`, `MP_LONG_LAST_FLAG`, `MP_LONG_SEQ_DIFF`, `MP_RECV_SEQ_DIFF`, `MP_SHORT_EXTEND`, `MP_SHORT_FIRST_FLAG`, `MP_SHORT_LAST_FLAG`, `MP_SHORT_SEQ_DIFF`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_NODE_PRIVATE`, `ng_ppp_crypt_rcv()`, `ng_ppp_frag_process()`, `NGI_GET_M`, `PROT_MP`, `ng_ppp_link_stat::runts`, `ng_ppp_link::seq`, `ng_ppp_frag::seq`, `ng_ppp_link::stats`, and `ng_ppp_frag::timestamp`.

Referenced by `ng_ppp_rcvdata()`.

Here is the call graph for this function:



7.131.3.24 `static void ng_ppp_mp_strategy (node_p node, int len, int * distrib)` [static]

Definition at line 2068 of file `ng_ppp.c`.

References `NG_NODE_PRIVATE`, and `NG_PPP_MAX_LINKS`.

Referenced by `ng_ppp_mp_xmit()`.

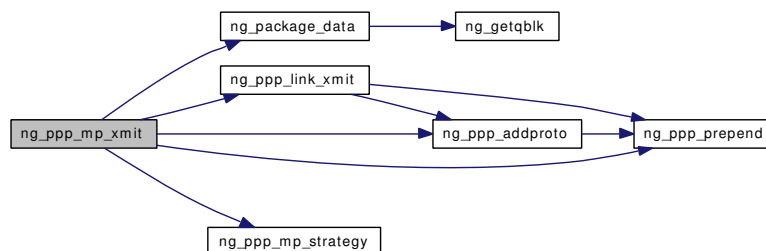
7.131.3.25 `static int ng_ppp_mp_xmit (node_p node, item_p item, uint16_t proto)` [static]

Definition at line 1846 of file `ng_ppp.c`.

References `ng_ppp_link::conf`, `MP_LONG_FIRST_FLAG`, `MP_LONG_LAST_FLAG`, `MP_LONG_SEQ_MASK`, `MP_MIN_FRAG_LEN`, `MP_SHORT_FIRST_FLAG`, `MP_SHORT_LAST_FLAG`, `MP_SHORT_SEQ_MASK`, `ng_ppp_link_conf::mru`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_NODE_PRIVATE`, `NG_NOFLAGS`, `ng_package_data()`, `ng_ppp_addproto()`, `ng_ppp_link_xmit()`, `NG_PPP_MAX_LINKS`, `ng_ppp_mp_strategy()`, `ng_ppp_prepend()`, `NGI_GET_M`, `NGI_M`, and `PROT_MP`.

Referenced by `ng_ppp_crypt_xmit()`, and `ng_ppp_rcvdata_encrypt()`.

Here is the call graph for this function:



7.131.3.26 `static int ng_ppp_newhook (node_p node, hook_p hook, const char * name)`
 [static]

Definition at line 482 of file ng_ppp.c.

References `fn`, `NG_HOOK_FORCE_WRITER`, `NG_HOOK_SET_PRIVATE`, `NG_HOOK_SET_RCVDATA`, `NG_NODE_PRIVATE`, `NG_PPP_HOOK_LINK_PREFIX`, `ng_ppp_hook_names`, `NG_PPP_MAX_LINKS`, and `ng_ppp_update()`.

Here is the call graph for this function:



7.131.3.27 `static struct mbuf * ng_ppp_prepend (struct mbuf * m, const void * buf, int len)`
 [static]

Definition at line 2264 of file ng_ppp.c.

Referenced by `ng_ppp_addproto()`, `ng_ppp_bypass()`, `ng_ppp_link_xmit()`, and `ng_ppp_mp_xmit()`.

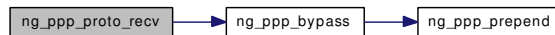
7.131.3.28 `static int ng_ppp_proto_rcv (node_p node, item_p item, uint16_t proto, uint16_t linkNum)` [static]

Definition at line 843 of file ng_ppp.c.

References `HOOK_INDEX_ATALK`, `HOOK_INDEX_INET`, `HOOK_INDEX_IPV6`, `HOOK_INDEX_IPX`, `NG_FWD_ITEM_HOOK`, `NG_NODE_PRIVATE`, `ng_ppp_bypass()`, `PROT_ATALK`, `PROT_IP`, `PROT_IPV6`, and `PROT_IPX`.

Referenced by `ng_ppp_hcomp_rcv()`, and `ng_ppp_rcvdata_vjc_ip()`.

Here is the call graph for this function:

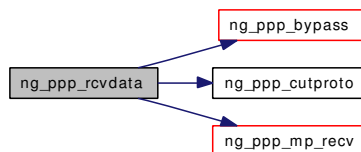


7.131.3.29 `static int ng_ppp_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 1303 of file ng_ppp.c.

References `ng_ppp_link_stat::badProtos`, `ng_ppp_link::conf`, `ng_ppp_link_conf::enableLink`, `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `ng_ppp_acf`, `ng_ppp_bypass()`, `ng_ppp_cutproto()`, `NG_PPP_MAX_LINKS`, `ng_ppp_mp_rcv()`, `NGI_GET_M`, `NGI_M`, `PROT_VALID`, `ng_ppp_link_stat::rcvFrames`, `ng_ppp_link_stat::rcvOctets`, and `ng_ppp_link::stats`.

Here is the call graph for this function:

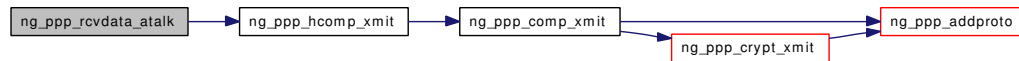


7.131.3.30 static int ng_ppp_rcvdata_atalk (hook_p hook, item_p item) [static]

Definition at line 754 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_hcomp_xmit(), and PROT_ATALK.

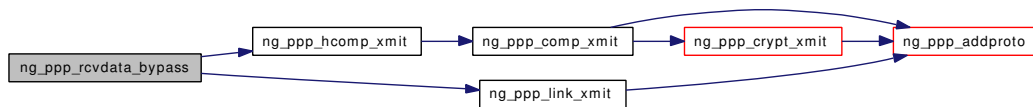
Here is the call graph for this function:

**7.131.3.31 static int ng_ppp_rcvdata_bypass (hook_p hook, item_p item) [static]**

Definition at line 786 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_PPP_BUNDLE_LINKNUM, ng_ppp_hcomp_xmit(), ng_ppp_link_xmit(), NGI_GET_M, and NGI_M.

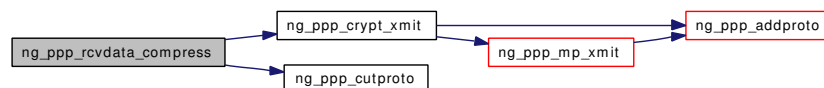
Here is the call graph for this function:

**7.131.3.32 static int ng_ppp_rcvdata_compress (hook_p hook, item_p item) [static]**

Definition at line 1027 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, NG_PPP_COMPRESS_FULL, NG_PPP_COMPRESS_NONE, ng_ppp_crypt_xmit(), ng_ppp_cutproto(), NGI_GET_M, NGI_M, PROT_COMPD, and PROT_VALID.

Here is the call graph for this function:

**7.131.3.33 static int ng_ppp_rcvdata_decompress (hook_p hook, item_p item) [static]**

Definition at line 1098 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, NG_PPP_BUNDLE_LINKNUM, ng_ppp_cutproto(), ng_ppp_hcomp_rcv(), NGI_GET_M, NGI_M, and PROT_VALID.

Here is the call graph for this function:

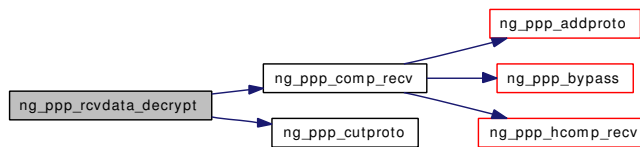


7.131.3.34 static int ng_ppp_rcvdata_decrypt (hook_p hook, item_p item) [static]

Definition at line 1202 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, NG_PPP_BUNDLE_LINKNUM, ng_ppp_comp_rcv(), ng_ppp_cutproto(), NGI_GET_M, NGI_M, and PROT_VALID.

Here is the call graph for this function:

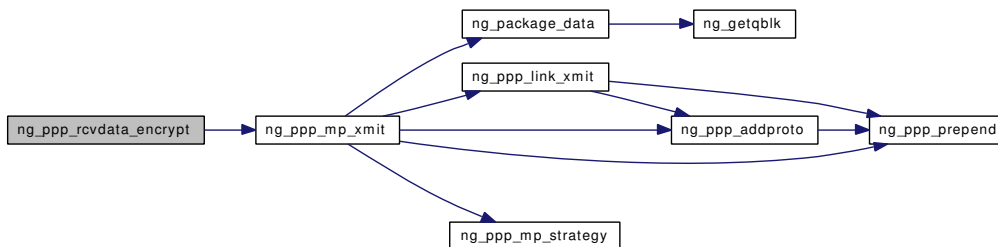


7.131.3.35 static int ng_ppp_rcvdata_encrypt (hook_p hook, item_p item) [static]

Definition at line 1158 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_mp_xmit(), and PROT_CRYPTD.

Here is the call graph for this function:

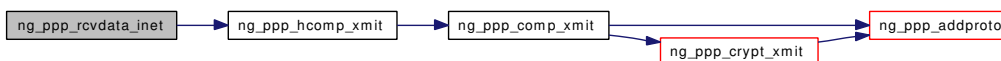


7.131.3.36 static int ng_ppp_rcvdata_inet (hook_p hook, item_p item) [static]

Definition at line 722 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_hcomp_xmit(), and PROT_IP.

Here is the call graph for this function:

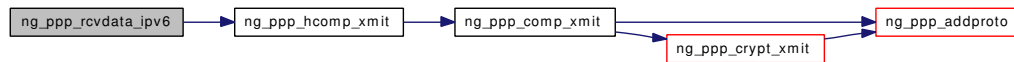


7.131.3.37 static int ng_ppp_rcvdata_ipv6 (hook_p hook, item_p item) [static]

Definition at line 738 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_hcomp_xmit(), and PROT_IPV6.

Here is the call graph for this function:

**7.131.3.38 static int ng_ppp_rcvdata_ipx (hook_p hook, item_p item) [static]**

Definition at line 770 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_hcomp_xmit(), and PROT_IPX.

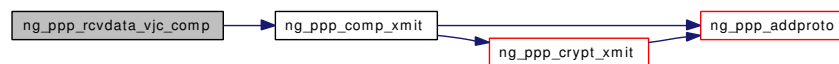
Here is the call graph for this function:

**7.131.3.39 static int ng_ppp_rcvdata_vjc_comp (hook_p hook, item_p item) [static]**

Definition at line 902 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ppp_comp_xmit(), and PROT_VJCOMP.

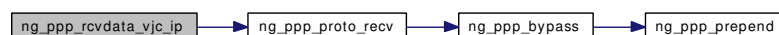
Here is the call graph for this function:

**7.131.3.40 static int ng_ppp_rcvdata_vjc_ip (hook_p hook, item_p item) [static]**

Definition at line 979 of file ng_ppp.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, NG_PPP_BUNDLE_LINKNUM, ng_ppp_proto_recv(), and PROT_IP.

Here is the call graph for this function:

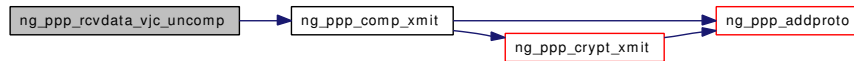


7.131.3.41 `static int ng_ppp_rcvdata_vjc_uncomp (hook_p hook, item_p item)` [static]

Definition at line 918 of file `ng_ppp.c`.

References `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `ng_ppp_comp_xmit()`, and `PROT_VJUNCOMP`.

Here is the call graph for this function:

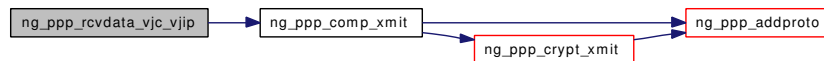


7.131.3.42 `static int ng_ppp_rcvdata_vjc_vjip (hook_p hook, item_p item)` [static]

Definition at line 934 of file `ng_ppp.c`.

References `NG_FREE_ITEM`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `ng_ppp_comp_xmit()`, and `PROT_IP`.

Here is the call graph for this function:

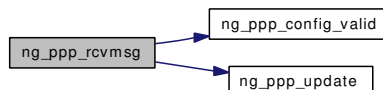


7.131.3.43 `static int ng_ppp_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 548 of file `ng_ppp.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ERROUT`, `ng_mesg::header`, `HOOK_INDEX_VJC_IP`, `MP_NOSEQ`, `NG_FREE_MSG`, `NG_FWD_ITEM_HOOK`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_PPP_BUNDLE_LINKNUM`, `ng_ppp_config_valid()`, `NG_PPP_MAX_LINKS`, `ng_ppp_update()`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGI_MSG`, `NGM_PPP_CLR_LINK_STATS`, `NGM_PPP_COOKIE`, `NGM_PPP_GET_CONFIG`, `NGM_PPP_GET_LINK_STATS`, `NGM_PPP_GET_MP_STATE`, `NGM_PPP_GETCLR_LINK_STATS`, `NGM_PPP_SET_CONFIG`, `NGM_VJC_COOKIE`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:

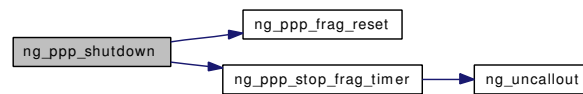


7.131.3.44 `static int ng_ppp_shutdown (node_p node)` [static]

Definition at line 673 of file `ng_ppp.c`.

References `M_NETGRAPH_PPP`, `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `ng_ppp_frag_reset()`, and `ng_ppp_stop_frag_timer()`.

Here is the call graph for this function:



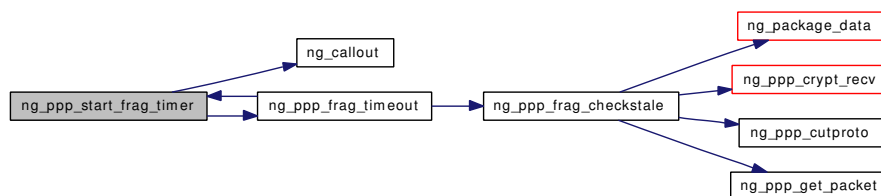
7.131.3.45 static void ng_ppp_start_frag_timer (node_p node) [static]

Definition at line 2419 of file ng_ppp.c.

References MP_FRAGTIMER_INTERVAL, ng_callout(), NG_NODE_PRIVATE, and ng_ppp_frag_timeout().

Referenced by ng_ppp_frag_timeout().

Here is the call graph for this function:



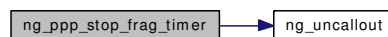
7.131.3.46 static void ng_ppp_stop_frag_timer (node_p node) [static]

Definition at line 2432 of file ng_ppp.c.

References NG_NODE_PRIVATE, and ng_unccallout().

Referenced by ng_ppp_shutdown().

Here is the call graph for this function:



7.131.3.47 static void ng_ppp_update (node_p node, int newConf) [static]

Definition at line 2277 of file ng_ppp.c.

References HOOK_INDEX_VJC_COMP, HOOK_INDEX_VJC_IP, HOOK_INDEX_VJC_UNCOMP, HOOK_INDEX_VJC_VJIP, MP_INITIAL_SEQ, MP_NOSEQ, NG_NODE_PRIVATE, and NG_PPP_MAX_LINKS.

Referenced by ng_ppp_disconnect(), ng_ppp_newhook(), and ng_ppp_rcvmsg().

7.131.4 Variable Documentation

7.131.4.1 `ng_rcvdata_t* fn`

Definition at line 252 of file `ng_ppp.c`.

Referenced by `ng_ppp_newhook()`.

7.131.4.2 `char* const name`

Definition at line 251 of file `ng_ppp.c`.

7.131.4.3 `const uint8_t ng_ppp_acf[2] = { 0xff, 0x03 }` [static]

Definition at line 441 of file `ng_ppp.c`.

Referenced by `ng_ppp_link_xmit()`, and `ng_ppp_rcvdata()`.

7.131.4.4 `struct ng_parse_fixedarray_info ng_ppp_array_info` [static]

Initial value:

```
{
    &ng_ppp_link_type,
    NG_PPP_MAX_LINKS
}
```

Definition at line 356 of file `ng_ppp.c`.

7.131.4.5 `struct ng_parse_type ng_ppp_bund_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_ppp_bund_type_fields
}
```

Definition at line 350 of file `ng_ppp.c`.

7.131.4.6 `struct ng_parse_struct_field ng_ppp_bund_type_fields[] = NG_PPP_BUND_TYPE_INFO` [static]

Definition at line 349 of file `ng_ppp.c`.

7.131.4.7 `struct ng_cmdlist ng_ppp_cmds[]` [static]

Definition at line 380 of file `ng_ppp.c`.

7.131.4.8 `struct ng_parse_type ng_ppp_conf_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_ppp_conf_type_fields
}
```

Definition at line 366 of file ng_ppp.c.

7.131.4.9 `struct ng_parse_struct_field ng_ppp_conf_type_fields[] = NG_PPP_CONFIG_TYPE_INFO(&ng_ppp_bund_type, &ng_ppp_link_array_type)` [static]

Definition at line 365 of file ng_ppp.c.

7.131.4.10 `ng_constructor_t ng_ppp_constructor` [static]

Definition at line 225 of file ng_ppp.c.

7.131.4.11 `ng_disconnect_t ng_ppp_disconnect` [static]

Definition at line 230 of file ng_ppp.c.

7.131.4.12 `struct { ... } ng_ppp_hook_names[]` [static]

Referenced by ng_ppp_newhook().

7.131.4.13 `struct ng_parse_type ng_ppp_link_array_type` [static]**Initial value:**

```
{
    &ng_parse_fixedarray_type,
    &ng_ppp_array_info,
}
```

Definition at line 360 of file ng_ppp.c.

7.131.4.14 `struct ng_parse_type ng_ppp_link_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_ppp_link_type_fields
}
```

Definition at line 342 of file ng_ppp.c.

7.131.4.15 `struct ng_parse_struct_field ng_ppp_link_type_fields[] =
NG_PPP_LINK_TYPE_INFO [static]`

Definition at line 341 of file ng_ppp.c.

7.131.4.16 `struct timeval ng_ppp_max_staleness = { 2, 0 } [static]`

Definition at line 444 of file ng_ppp.c.

7.131.4.17 `struct ng_parse_type ng_ppp_mp_state_type [static]`

Initial value:

```
{  
    &ng_parse_struct_type,  
    &ng_ppp_mp_state_type_fields  
}
```

Definition at line 334 of file ng_ppp.c.

7.131.4.18 `struct ng_parse_struct_field ng_ppp_mp_state_type_fields[] =
NG_PPP_MP_STATE_TYPE_INFO(&ng_ppp_rseq_array_type) [static]`

Definition at line 333 of file ng_ppp.c.

7.131.4.19 `ng_newhook_t ng_ppp_newhook [static]`

Definition at line 228 of file ng_ppp.c.

7.131.4.20 `ng_rcvdata_t ng_ppp_rcvdata [static]`

Definition at line 229 of file ng_ppp.c.

7.131.4.21 `ng_rcvdata_t ng_ppp_rcvdata_atalk [static]`

Definition at line 235 of file ng_ppp.c.

7.131.4.22 `ng_rcvdata_t ng_ppp_rcvdata_bypass [static]`

Definition at line 236 of file ng_ppp.c.

7.131.4.23 `ng_rcvdata_t ng_ppp_rcvdata_compress [static]`

Definition at line 243 of file ng_ppp.c.

7.131.4.24 `ng_rcvdata_t ng_ppp_rcvdata_decompress [static]`

Definition at line 244 of file ng_ppp.c.

7.131.4.25 [ng_rcvdata_t ng_ppp_rcvdata_decrypt](#) [static]

Definition at line 247 of file ng_ppp.c.

7.131.4.26 [ng_rcvdata_t ng_ppp_rcvdata_encrypt](#) [static]

Definition at line 246 of file ng_ppp.c.

7.131.4.27 [ng_rcvdata_t ng_ppp_rcvdata_inet](#) [static]

Definition at line 232 of file ng_ppp.c.

7.131.4.28 [ng_rcvdata_t ng_ppp_rcvdata_ipv6](#) [static]

Definition at line 233 of file ng_ppp.c.

7.131.4.29 [ng_rcvdata_t ng_ppp_rcvdata_ipx](#) [static]

Definition at line 234 of file ng_ppp.c.

7.131.4.30 [ng_rcvdata_t ng_ppp_rcvdata_vjc_comp](#) [static]

Definition at line 239 of file ng_ppp.c.

7.131.4.31 [ng_rcvdata_t ng_ppp_rcvdata_vjc_ip](#) [static]

Definition at line 238 of file ng_ppp.c.

7.131.4.32 [ng_rcvdata_t ng_ppp_rcvdata_vjc_uncomp](#) [static]

Definition at line 240 of file ng_ppp.c.

7.131.4.33 [ng_rcvdata_t ng_ppp_rcvdata_vjc_vjip](#) [static]

Definition at line 241 of file ng_ppp.c.

7.131.4.34 [ng_rcvmsg_t ng_ppp_rcvmsg](#) [static]

Definition at line 226 of file ng_ppp.c.

7.131.4.35 [struct ng_parse_fixedarray_info ng_ppp_rseq_array_info](#) [static]**Initial value:**

```
{
    &ng_parse_hint32_type,
    NG_PPP_MAX_LINKS
}
```

Definition at line 324 of file ng_ppp.c.

7.131.4.36 `struct ng_parse_type ng_ppp_rseq_array_type` [static]

Initial value:

```
{
    &ng_parse_fixedarray_type,
    &ng_ppp_rseq_array_info,
}
```

Definition at line 328 of file ng_ppp.c.

7.131.4.37 `ng_shutdown_t ng_ppp_shutdown` [static]

Definition at line 227 of file ng_ppp.c.

7.131.4.38 `struct ng_parse_type ng_ppp_stats_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_ppp_stats_type_fields
}
```

Definition at line 374 of file ng_ppp.c.

7.131.4.39 `struct ng_parse_struct_field ng_ppp_stats_type_fields[] =
NG_PPP_STATS_TYPE_INFO` [static]

Definition at line 373 of file ng_ppp.c.

7.131.4.40 `struct ng_type ng_ppp_typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_PPP_NODE_TYPE,
    .constructor = ng_ppp_constructor,
    .rcvmsg =      ng_ppp_rcvmsg,
    .shutdown =    ng_ppp_shutdown,
    .newhook =     ng_ppp_newhook,
    .rcvdata =     ng_ppp_rcvdata,
    .disconnect =  ng_ppp_disconnect,
    .cmdlist =     ng_ppp_cmds,
}
```

Definition at line 427 of file ng_ppp.c.

7.132 /usr/src/sys/netgraph/ng_ppp.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_ppp_mp_state](#)
- struct [ng_ppp_link_conf](#)
- struct [ng_ppp_bund_conf](#)
- struct [ng_ppp_node_conf](#)
- struct [ng_ppp_link_stat](#)

Defines

- #define [NG_PPP_NODE_TYPE](#) "ppp"
- #define [NGM_PPP_COOKIE](#) 940897795
- #define [NG_PPP_MAX_LINKS](#) 16
- #define [NG_PPP_BUNDLE_LINKNUM](#) 0xffff
- #define [NG_PPP_MAX_LATENCY](#) 1000
- #define [NG_PPP_MAX_BANDWIDTH](#) 125000
- #define [NG_PPP_HOOK_BYPASS](#) "bypass"
- #define [NG_PPP_HOOK_COMPRESS](#) "compress"
- #define [NG_PPP_HOOK_DECOMPRESS](#) "decompress"
- #define [NG_PPP_HOOK_ENCRYPT](#) "encrypt"
- #define [NG_PPP_HOOK_DECRYPT](#) "decrypt"
- #define [NG_PPP_HOOK_VJC_IP](#) "vjc_ip"
- #define [NG_PPP_HOOK_VJC_COMP](#) "vjc_vjcomp"
- #define [NG_PPP_HOOK_VJC_UNCOMP](#) "vjc_vjuncomp"
- #define [NG_PPP_HOOK_VJC_VJIP](#) "vjc_vjip"
- #define [NG_PPP_HOOK_INET](#) "inet"
- #define [NG_PPP_HOOK_ATALK](#) "atalk"
- #define [NG_PPP_HOOK_IPX](#) "ipx"
- #define [NG_PPP_HOOK_IPV6](#) "ipv6"
- #define [NG_PPP_HOOK_LINK_PREFIX](#) "link"
- #define [NG_PPP_MP_STATE_TYPE_INFO](#)(atype)
- #define [NG_PPP_LINK_TYPE_INFO](#)
- #define [NG_PPP_BUND_TYPE_INFO](#)
- #define [NG_PPP_CONFIG_TYPE_INFO](#)(bctype, arctype)
- #define [NG_PPP_STATS_TYPE_INFO](#)

Enumerations

- enum { NG_PPP_COMPRESS_NONE = 0, NG_PPP_COMPRESS_SIMPLE, NG_PPP_COMPRESS_FULL }
- enum { NG_PPP_DECOMPRESS_NONE = 0, NG_PPP_DECOMPRESS_SIMPLE, NG_PPP_DECOMPRESS_FULL }
- enum {
 NGM_PPP_SET_CONFIG = 1, NGM_PPP_GET_CONFIG, NGM_PPP_GET_MP_STATE,
 NGM_PPP_GET_LINK_STATS,
 NGM_PPP_CLR_LINK_STATS, NGM_PPP_GETCLR_LINK_STATS }

7.132.1 Define Documentation

7.132.1.1 #define NG_PPP_BUND_TYPE_INFO

Value:

```
{
    { "mrru",                &ng_parse_uint16_type }, \
    { "enableMultilink",    &ng_parse_uint8_type  }, \
    { "recvShortSeq",      &ng_parse_uint8_type  }, \
    { "xmitShortSeq",      &ng_parse_uint8_type  }, \
    { "enableRoundRobin",  &ng_parse_uint8_type  }, \
    { "enableIP",          &ng_parse_uint8_type  }, \
    { "enableIPv6",        &ng_parse_uint8_type  }, \
    { "enableAtalk",       &ng_parse_uint8_type  }, \
    { "enableIPX",         &ng_parse_uint8_type  }, \
    { "enableCompression", &ng_parse_uint8_type  }, \
    { "enableDecompression", &ng_parse_uint8_type }, \
    { "enableEncryption",  &ng_parse_uint8_type  }, \
    { "enableDecryption",  &ng_parse_uint8_type  }, \
    { "enableVJCompression", &ng_parse_uint8_type }, \
    { "enableVJDecompression", &ng_parse_uint8_type }, \
    { NULL }
}
```

Definition at line 158 of file ng_ppp.h.

7.132.1.2 #define NG_PPP_BUNDLE_LINKNUM 0xffff

Definition at line 55 of file ng_ppp.h.

Referenced by ng_ppp_frag_checkstale(), ng_ppp_frag_process(), ng_ppp_rcvdata_bypass(), ng_ppp_rcvdata_decompress(), ng_ppp_rcvdata_decrypt(), ng_ppp_rcvdata_vjc_ip(), and ng_ppp_rcvmsg().

7.132.1.3 #define NG_PPP_CONFIG_TYPE_INFO(bctype, arytype)

Value:

```
{
    \
    { "bund",                (bctype)      }, \
    { "links",              (arytype)    }, \
    { NULL }
}
```

Definition at line 184 of file ng_ppp.h.

7.132.1.4 #define NG_PPP_HOOK_ATALK "atalk"

Definition at line 72 of file ng_ppp.h.

7.132.1.5 #define NG_PPP_HOOK_BYPASS "bypass"

Definition at line 62 of file ng_ppp.h.

7.132.1.6 #define NG_PPP_HOOK_COMPRESS "compress"

Definition at line 63 of file ng_ppp.h.

7.132.1.7 #define NG_PPP_HOOK_DECOMPRESS "decompress"

Definition at line 64 of file ng_ppp.h.

7.132.1.8 #define NG_PPP_HOOK_DECRYPT "decrypt"

Definition at line 66 of file ng_ppp.h.

7.132.1.9 #define NG_PPP_HOOK_ENCRYPT "encrypt"

Definition at line 65 of file ng_ppp.h.

7.132.1.10 #define NG_PPP_HOOK_INET "inet"

Definition at line 71 of file ng_ppp.h.

7.132.1.11 #define NG_PPP_HOOK_IPV6 "ipv6"

Definition at line 74 of file ng_ppp.h.

7.132.1.12 #define NG_PPP_HOOK_IPX "ipx"

Definition at line 73 of file ng_ppp.h.

7.132.1.13 #define NG_PPP_HOOK_LINK_PREFIX "link"

Definition at line 76 of file ng_ppp.h.

Referenced by ng_ppp_newhook().

7.132.1.14 #define NG_PPP_HOOK_VJC_COMP "vjc_vjcomp"

Definition at line 68 of file ng_ppp.h.

7.132.1.15 #define NG_PPP_HOOK_VJC_IP "vjc_ip"

Definition at line 67 of file ng_ppp.h.

7.132.1.16 #define NG_PPP_HOOK_VJC_UNCOMP "vjc_vjuncomp"

Definition at line 69 of file ng_ppp.h.

7.132.1.17 #define NG_PPP_HOOK_VJC_VJIP "vjc_vjip"

Definition at line 70 of file ng_ppp.h.

7.132.1.18 #define NG_PPP_LINK_TYPE_INFO**Value:**

```
{
    { "enableLink",          &ng_parse_uint8_type }, \
    { "enableProtoComp",   &ng_parse_uint8_type }, \
    { "enableACFComp",     &ng_parse_uint8_type }, \
    { "mru",                &ng_parse_uint16_type }, \
    { "latency",           &ng_parse_uint32_type }, \
    { "bandwidth",        &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 128 of file ng_ppp.h.

7.132.1.19 #define NG_PPP_MAX_BANDWIDTH 125000

Definition at line 59 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

7.132.1.20 #define NG_PPP_MAX_LATENCY 1000

Definition at line 58 of file ng_ppp.h.

Referenced by ng_ppp_config_valid().

7.132.1.21 #define NG_PPP_MAX_LINKS 16

Definition at line 52 of file ng_ppp.h.

Referenced by ng_ppp_config_valid(), ng_ppp_constructor(), ng_ppp_link_xmit(), ng_ppp_mp_strategy(), ng_ppp_mp_xmit(), ng_ppp_newhook(), ng_ppp_rcvdata(), ng_ppp_rcvmsg(), and ng_ppp_update().

7.132.1.22 #define NG_PPP_MP_STATE_TYPE_INFO(atype)**Value:**

```

{
    \
    { "rseq",      (atype)          },      \
    { "mseq",     &ng_parse_hint32_type },      \
    { "xseq",     &ng_parse_hint32_type },      \
    { NULL }
}

```

Definition at line 110 of file ng_ppp.h.

7.132.1.23 #define NG_PPP_NODE_TYPE "ppp"

Definition at line 48 of file ng_ppp.h.

7.132.1.24 #define NG_PPP_STATS_TYPE_INFO

Value:

```

{
    \
    { "xmitFrames",    &ng_parse_uint32_type },      \
    { "xmitOctets",   &ng_parse_uint32_type },      \
    { "recvFrames",   &ng_parse_uint32_type },      \
    { "recvOctets",   &ng_parse_uint32_type },      \
    { "badProtos",    &ng_parse_uint32_type },      \
    { "runts",        &ng_parse_uint32_type },      \
    { "dupFragments", &ng_parse_uint32_type },      \
    { "dropFragments", &ng_parse_uint32_type },      \
    { NULL }
}

```

Definition at line 203 of file ng_ppp.h.

7.132.1.25 #define NGM_PPP_COOKIE 940897795

Definition at line 49 of file ng_ppp.h.

Referenced by ng_ppp_rcvmsg().

7.132.2 Enumeration Type Documentation

7.132.2.1 anonymous enum

Enumerator:

NG_PPP_COMPRESS_NONE
NG_PPP_COMPRESS_SIMPLE
NG_PPP_COMPRESS_FULL

Definition at line 79 of file ng_ppp.h.

7.132.2.2 anonymous enum

Enumerator:

NG_PPP_DECOMPRESS_NONE

NG_PPP_DECOMPRESS_SIMPLE
NG_PPP_DECOMPRESS_FULL

Definition at line 86 of file ng_ppp.h.

7.132.2.3 anonymous enum

Enumerator:

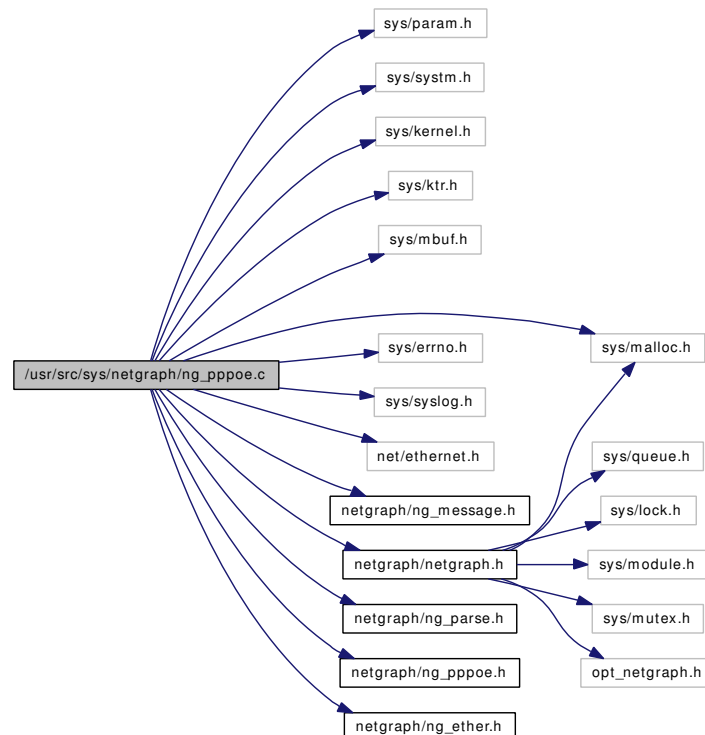
NGM_PPP_SET_CONFIG
NGM_PPP_GET_CONFIG
NGM_PPP_GET_MP_STATE
NGM_PPP_GET_LINK_STATS
NGM_PPP_CLR_LINK_STATS
NGM_PPP_GETCLR_LINK_STATS

Definition at line 93 of file ng_ppp.h.

7.133 /usr/src/sys/netgraph/ng_pppoe.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/ktr.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <net/ethernet.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_pppoe.h>
#include <netgraph/ng_ether.h>
```

Include dependency graph for ng_pppoe.c:



Data Structures

- struct [sess_neg](#)

- struct [sess_con](#)
- struct [PPPoE](#)
- union [uniq](#)

Defines

- #define [M_NETGRAPH_PPPOE](#) M_NETGRAPH
- #define [SIGNOFF](#) "session closed"
- #define [OFFSETOF](#)(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))
- #define [NUMTAGS](#) 20
- #define [NG_PPPOE_SESSION_NODE](#)(sp) NG_HOOK_NODE(sp → hook)
- #define [COMPAT_3COM](#) 0x00000001
- #define [COMPAT_DLINK](#) 0x00000002
- #define [LEAVE](#)(x) do { error = x; goto quit; } while(0)

Typedefs

- typedef [sess_neg](#) * [negp](#)
- typedef [sess_con](#) * [sessp](#)
- typedef [PPPoE](#) * [priv_p](#)

Enumerations

- enum [state](#) {
[PPPOE_SNONE](#) = 0, [PPPOE_LISTENING](#), [PPPOE_SINIT](#), [PPPOE_PRIMED](#),
[PPPOE_SOFFER](#), [PPPOE_SREQ](#), [PPPOE_NEWCONNECTED](#), [PPPOE_CONNECTED](#),
[PPPOE_DEAD](#) }

Functions

- [NETGRAPH_INIT](#) (pppoe,&typestruct)
- static void [pppoe_start](#) (sessp sp)
- static void [ng_pppoe_sendpacket](#) (sessp sp)
- static void [pppoe_ticker](#) (node_p node, hook_p hook, void *arg1, int arg2)
- static struct [pppoe_tag](#) * [scan_tags](#) (sessp sp, const struct [pppoe_hdr](#) *ph)
- static int [pppoe_send_event](#) (sessp sp, enum [cmd](#) cmdid)
- static uint16_t [get_new_sid](#) (node_p node)
- static __inline struct [pppoe_tag](#) * [next_tag](#) (const struct [pppoe_hdr](#) *ph)
- static struct [pppoe_tag](#) * [get_tag](#) (const struct [pppoe_hdr](#) *ph, uint16_t idx)
- static void [init_tags](#) (sessp sp)
- static void [insert_tag](#) (sessp sp, const struct [pppoe_tag](#) *tp)
- static void [make_packet](#) (sessp sp)
- static hook_p [pppoe_match_svc](#) (node_p node, const struct [pppoe_tag](#) *tag)
- static int [pppoe_broadcast_padi](#) (node_p node, struct mbuf *m0)
- static hook_p [pppoe_find_svc](#) (node_p node, const char *svc_name, int svc_len)
- static hook_p [pppoe_findsession](#) (node_p node, const struct [pppoe_full_hdr](#) *wh)
- static hook_p [pppoe_finduniq](#) (node_p node, const struct [pppoe_tag](#) *tag)
- static int [ng_pppoe_constructor](#) (node_p node)

- static int `ng_pppoe_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_pppoe_connect` (`hook_p` hook)
- static int `ng_pppoe_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `send_acname` (`sessp` sp, const struct `pppoe_tag` *tag)
- static int `send_sessionid` (`sessp` sp)
- static int `ng_pppoe_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_pppoe_shutdown` (`node_p` node)
- static int `ng_pppoe_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_pppoe_constructor`
- static `ng_rcvmsg_t` `ng_pppoe_rcvmsg`
- static `ng_shutdown_t` `ng_pppoe_shutdown`
- static `ng_newhook_t` `ng_pppoe_newhook`
- static `ng_connect_t` `ng_pppoe_connect`
- static `ng_rcvdata_t` `ng_pppoe_rcvdata`
- static `ng_disconnect_t` `ng_pppoe_disconnect`
- static struct `ng_parse_struct_field` `ngpppoe_init_data_type_fields` [] = NG_PPPOE_INIT_DATA_TYPE_INFO
- static struct `ng_parse_type` `ngpppoe_init_data_state_type`
- static struct `ng_parse_struct_field` `ng_pppoe_sts_type_fields` [] = NG_PPPOE_STS_TYPE_INFO
- static struct `ng_parse_type` `ng_pppoe_sts_state_type`
- static struct `ng_cmdlist` `ng_pppoe_cmds` []
- static struct `ng_type` `typestruct`

7.133.1 Define Documentation

7.133.1.1 #define COMPAT_3COM 0x00000001

Definition at line 252 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_rcvdata()`, and `ng_pppoe_rcvmsg()`.

7.133.1.2 #define COMPAT_DLINK 0x00000002

Definition at line 253 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_rcvdata()`, and `ng_pppoe_rcvmsg()`.

7.133.1.3 #define LEAVE(x) do { error = x; goto quit; } while(0)

Definition at line 263 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_rcvdata()`, and `ng_pppoe_rcvmsg()`.

7.133.1.4 #define M_NETGRAPH_PPPOE M_NETGRAPH

Definition at line 63 of file `ng_pppoe.c`.

Referenced by `ng_pppoe_constructor()`, `ng_pppoe_disconnect()`, `ng_pppoe_newhook()`, `ng_pppoe_rcvdata()`, `ng_pppoe_rcvmsg()`, and `ng_pppoe_shutdown()`.

7.133.1.5 #define NG_PPPOE_SESSION_NODE(sp) NG_HOOK_NODE(sp → hook)

Definition at line 240 of file ng_pppoe.c.

Referenced by pppoe_start().

7.133.1.6 #define NUMTAGS 20

Definition at line 204 of file ng_pppoe.c.

Referenced by insert_tag(), and make_packet().

7.133.1.7 #define OFFSETOF(s, e) ((char *)&((s *)0) → e - (char *)&((s *)0))

Definition at line 67 of file ng_pppoe.c.

7.133.1.8 #define SIGNOFF "session closed"

Definition at line 66 of file ng_pppoe.c.

Referenced by ng_pppoe_disconnect().

7.133.2 Typedef Documentation**7.133.2.1 typedef struct sess_neg* negp**

Definition at line 225 of file ng_pppoe.c.

7.133.2.2 typedef struct PPPoE* priv_p

Definition at line 256 of file ng_pppoe.c.

7.133.2.3 typedef struct sess_con* sessp

Definition at line 238 of file ng_pppoe.c.

7.133.3 Enumeration Type Documentation**7.133.3.1 enum state**

Enumerator:

PPPOE_SNONE

PPPOE_LISTENING

PPPOE_SINIT

PPPOE_PRIMED

PPPOE_SOFFER

PPPOE_SREQ

PPPOE_NEWCONNECTED

PPPOE_CONNECTED***PPPOE_DEAD***

Definition at line 192 of file ng_pppoe.c.

7.133.4 Function Documentation

7.133.4.1 static uint16_t get_new_sid (node_p node) [static]

Definition at line 281 of file ng_pppoe.c.

References NG_HOOK_PRIVATE, NG_NODE_PRIVATE, and sess_con::Session_ID.

Referenced by ng_pppoe_rcvdata().

7.133.4.2 static struct pppoe_tag* get_tag (const struct pppoe_hdr *ph, uint16_t idx) [static]

Definition at line 333 of file ng_pppoe.c.

References next_tag(), pppoe_hdr::tag, pppoe_tag::tag_len, and pppoe_tag::tag_type.

Referenced by ng_pppoe_rcvdata().

Here is the call graph for this function:



7.133.4.3 static void init_tags (sessp sp) [static]

Definition at line 371 of file ng_pppoe.c.

References sess_con::neg, and sess_neg::numtags.

Referenced by ng_pppoe_rcvdata(), and pppoe_start().

7.133.4.4 static void insert_tag (sessp sp, const struct pppoe_tag *tp) [static]

Definition at line 378 of file ng_pppoe.c.

References sess_con::neg, NUMTAGS, sess_neg::numtags, and sess_neg::tags.

Referenced by ng_pppoe_rcvdata(), pppoe_start(), and scan_tags().

7.133.4.5 static void make_packet (sessp sp) [static]

Definition at line 402 of file ng_pppoe.c.

References pppoe_hdr::length, sess_neg::m, sess_con::neg, sess_neg::numtags, NUMTAGS, pppoe_full_hdr::ph, sess_neg::pkt, packet::pkt_header, sess_con::Session_ID, pppoe_hdr::tag, and sess_neg::tags.

Referenced by ng_pppoe_rcvdata(), and pppoe_start().

7.133.4.6 NETGRAPH_INIT (pppoe, & typestruct)

7.133.4.7 static __inline struct pppoe_tag* next_tag (const struct pppoe_hdr *ph) [static]

Definition at line 321 of file ng_pppoe.c.

References pppoe_hdr::length, and pppoe_hdr::tag.

Referenced by get_tag(), and scan_tags().

7.133.4.8 static int ng_pppoe_connect (hook_p hook) [static]

Definition at line 692 of file ng_pppoe.c.

References NG_HOOK_NODE, NG_MKMESSAGE, NG_NODE_ID, NG_NODE_PRIVATE, NG_PEER_NODE, NG_SEND_MSG_ID, NGM_ETHER_COOKIE, and NGM_ETHER_GET_ENADDR.

7.133.4.9 static int ng_pppoe_constructor (node_p node) [static]

Definition at line 624 of file ng_pppoe.c.

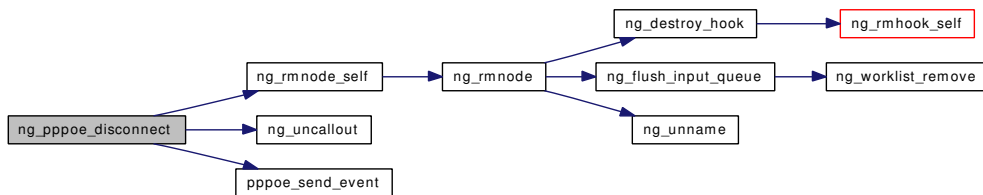
References ETHERTYPE_PPPOE_DISC, M_NETGRAPH_PPPOE, ng_node::nd_ID, and NG_NODE_SET_PRIVATE.

7.133.4.10 static int ng_pppoe_disconnect (hook_p hook) [static]

Definition at line 1677 of file ng_pppoe.c.

References pppoe_hdr::code, pppoe_full_hdr::eh, ETHERTYPE_PPPOE_3COM_DISC, ETHERTYPE_PPPOE_3COM_SESS, ETHERTYPE_PPPOE_DISC, sess_neg::handle, sess_neg::m, M_NETGRAPH_PPPOE, ng_node::nd_ID, sess_con::neg, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), NG_SEND_DATA_ONLY, ng_uncallout(), NGM_PPPOE_CLOSE, PADT_CODE, pppoe_full_hdr::ph, sess_con::pkt_hdr, PPPOE_CONNECTED, PPPOE_NEWCONNECTED, pppoe_send_event(), PPPOE_SNONE, PTT_GEN_ERR, SIGNOFF, sess_con::state, pppoe_tag::tag_data, pppoe_tag::tag_len, and pppoe_tag::tag_type.

Here is the call graph for this function:



7.133.4.11 static int ng_pppoe_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 657 of file ng_pppoe.c.

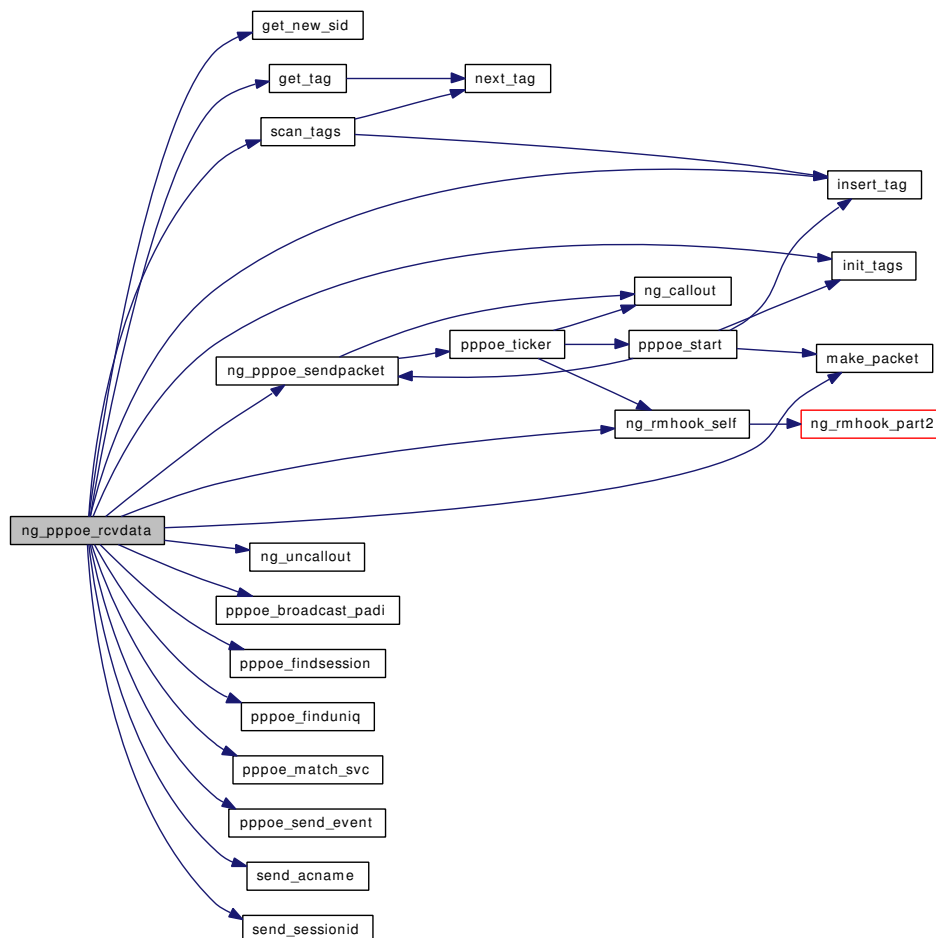
References M_NETGRAPH_PPPOE, ng_node::nd_ID, NG_HOOK_SET_PRIVATE, NG_NODE_PRIVATE, NG_PPPOE_HOOK_DEBUG, and NG_PPPOE_HOOK_ETHERNET.

7.133.4.12 static int ng_pppoe_rcvdata (hook_p hook, item_p item) [static]

Definition at line 1125 of file ng_pppoe.c.

References `__packed`, `sess_neg::ac_name`, `pppoe_hdr::code`, `COMPAT_3COM`, `COMPAT_DLINK`, `pppoe_full_hdr::eh`, `ETHERTYPE_PPPOE_3COM_DISC`, `ETHERTYPE_PPPOE_3COM_SESS`, `ETHERTYPE_PPPOE_DISC`, `ETHERTYPE_PPPOE_SESS`, `get_new_sid()`, `get_tag()`, `sess_neg::handle`, `datatag::hdr`, `ng_hook::hk_name`, `init_tags()`, `insert_tag()`, `LEAVE`, `pppoe_hdr::length`, `sess_neg::m`, `M_NETGRAPH_PPPOE`, `make_packet()`, `ng_node::nd_ID`, `sess_con::neg`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_ITEM_HOOK`, `NG_FWD_NEW_DATA`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `ng_pppoe_sendpacket()`, `ng_rmhook_self()`, `ng_uncallout()`, `NGI_GET_M`, `NGM_PPPOE_SUCCESS`, `PADI_CODE`, `PADO_CODE`, `PADR_CODE`, `PADS_CODE`, `PADT_CODE`, `pppoe_full_hdr::ph`, `sess_neg::pkt`, `sess_con::pkt_hdr`, `packet::pkt_header`, `pppoe_broadcast_padi()`, `PPPOE_CONNECTED`, `PPPOE_DEAD`, `pppoe_findsession()`, `pppoe_finduniq()`, `PPPOE_LISTENING`, `pppoe_match_svc()`, `PPPOE_NEWCONNECTED`, `PPPOE_PRIMED`, `pppoe_send_event()`, `PPPOE_SINIT`, `PPPOE_SNONE`, `PPPOE_SOFFER`, `PPPOE_SREQ`, `PTT_AC_COOKIE`, `PTT_AC_NAME`, `PTT_HOST_UNIQ`, `PTT_SRV_NAME`, `scan_tags()`, `send_acname()`, `send_sessionid()`, `sess_neg::service`, `pppoe_hdr::sid`, `sess_con::state`, `pppoe_tag::tag_len`, and `sess_neg::timeout`.

Here is the call graph for this function:

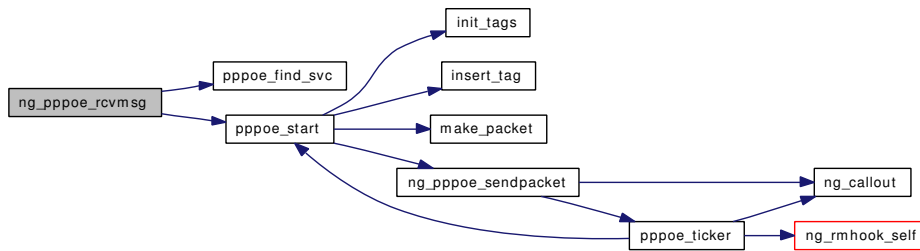


7.133.4.13 static int ng_pppoe_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 726 of file ng_pppoe.c.

References sess_neg::ac_name, sess_neg::ac_name_len, ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, pppoe_hdr::code, COMPAT_3COM, COMPAT_DLINK, sess_con::creator, datatag::data, ngpppoe_init_data::data, ng_mesg::data, ngpppoe_init_data::data_len, ETHERTYPE_PPPOE_3COM_DISC, ETHERTYPE_PPPOE_DISC, ng_mesg::ng_msghdr::flags, datatag::hdr, ng_mesg::header, ngpppoe_init_data::hook, LEAVE, M_NETGRAPH_PPPOE, ng_node::nd_ID, sess_con::neg, ng_callout_init, NG_FREE_MSG, NG_HOOK_NAME, NG_HOOK_PRIVATE, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_PPPOE_3COM, NG_PPPOE_DLINK, NG_PPPOE_STANDARD, NG_RESPOND_MSG, NGF_RESP, NGI_GET_MSG, NGI_RETADDR, NGM_ETHER_COOKIE, NGM_ETHER_GET_ENADDR, NGM_PPPOE_CONNECT, NGM_PPPOE_COOKIE, NGM_PPPOE_GET_STATUS, NGM_PPPOE_GETMODE, NGM_PPPOE_LISTEN, NGM_PPPOE_OFFER, NGM_PPPOE_SERVICE, NGM_PPPOE_SETENADDR, NGM_PPPOE_SETMODE, PADO_CODE, PADT_CODE, pppoe_full_hdr::ph, sess_neg::pkt, packet::pkt_header, pppoe_find_svc(), PPPOE_LISTENING, PPPOE_PRIMED, PPPOE_SERVICE_NAME_SIZE, PPPOE_SNONE, pppoe_start(), PTT_AC_NAME, PTT_SRV_NAME, sess_neg::service, sess_neg::service_len, sess_con::state, pppoe_tag::tag_len, pppoe_tag::tag_type, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



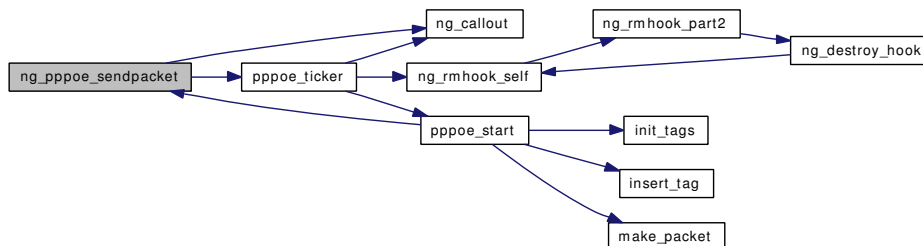
7.133.4.14 static void ng_pppoe_sendpacket (sessp sp) [static]

Definition at line 1827 of file ng_pppoe.c.

References sess_neg::handle, sess_con::hook, sess_neg::m, sess_con::neg, ng_callout(), NG_HOOK_NODE, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, PPPOE_CONNECTED, PPPOE_DEAD, PPPOE_INITIAL_TIMEOUT, PPPOE_LISTENING, PPPOE_NEWCONNECTED, PPPOE_OFFER_TIMEOUT, PPPOE_PRIMED, PPPOE_SINIT, PPPOE_SNONE, PPPOE_SOFFER, PPPOE_SREQ, pppoe_ticker(), sess_con::Session_ID, sess_con::state, and sess_neg::timeout.

Referenced by ng_pppoe_rcvdata(), and pppoe_start().

Here is the call graph for this function:



7.133.4.15 static int ng_pppoe_shutdown (node_p node) [static]

Definition at line 1660 of file ng_pppoe.c.

References M_NETGRAPH_PPPOE, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.133.4.16 static int pppoe_broadcast_padi (node_p node, struct mbuf * m0) [static]

Definition at line 495 of file ng_pppoe.c.

References NG_HOOK_PRIVATE, NG_NODE_PRIVATE, NG_SEND_DATA_ONLY, PPPOE_LISTENING, and sess_con::state.

Referenced by ng_pppoe_rcvdata().

7.133.4.17 static hook_p pppoe_find_svc (node_p node, const char * svc_name, int svc_len) [static]

Definition at line 531 of file ng_pppoe.c.

References datatag::data, sess_con::neg, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, PPPOE_LISTENING, sess_neg::service, sess_neg::service_len, and sess_con::state.

Referenced by ng_pppoe_rcvmsg().

7.133.4.18 static hook_p pppoe_findsession (node_p node, const struct pppoe_full_hdr * wh) [static]

Definition at line 563 of file ng_pppoe.c.

References pppoe_full_hdr::eh, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, pppoe_full_hdr::ph, sess_con::pkt_hdr, PPPOE_CONNECTED, PPPOE_NEWCONNECTED, sess_con::Session_ID, pppoe_hdr::sid, and sess_con::state.

Referenced by ng_pppoe_rcvdata().

7.133.4.19 static hook_p pppoe_finduniq (node_p node, const struct pppoe_tag * tag) [static]

Definition at line 595 of file ng_pppoe.c.

References uniq::bytes, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, uniq::pointer, and pppoe_tag::tag_data.

Referenced by ng_pppoe_rcvdata().

7.133.4.20 static hook_p pppoe_match_svc (node_p node, const struct pppoe_tag * tag) [static]

Definition at line 447 of file ng_pppoe.c.

References datatag::data, sess_con::neg, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, PPPOE_LISTENING, sess_neg::service, sess_neg::service_len, sess_con::state, pppoe_tag::tag_data, and pppoe_tag::tag_len.

Referenced by ng_pppoe_rcvdata().

7.133.4.21 `static int pppoe_send_event (sessp sp, enum cmd cmdid)` [static]

Definition at line 1931 of file `ng_pppoe.c`.

References `sess_con::creator`, `ng_mesg::data`, `sess_con::hook`, `ngpppoe_sts::hook`, `NG_HOOK_NAME`, `NG_HOOK_NODE`, `NG_HOOKSIZ`, `NG_MKMESSAGE`, `NG_SEND_MSG_ID`, `NGM_PPPOE_COOKIE`, and `sess_con::Session_ID`.

Referenced by `ng_pppoe_disconnect()`, and `ng_pppoe_rcvdata()`.

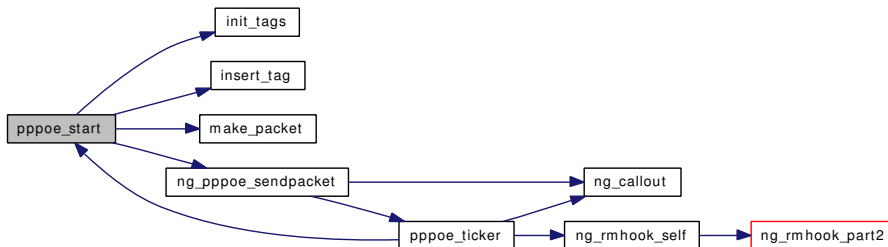
7.133.4.22 `static void pppoe_start (sessp sp)` [static]

Definition at line 1047 of file `ng_pppoe.c`.

References `__packed`, `pppoe_hdr::code`, `pppoe_full_hdr::eh`, `datatag::hdr`, `init_tags()`, `insert_tag()`, `make_packet()`, `sess_con::neg`, `NG_NODE_PRIVATE`, `ng_pppoe_sendpacket()`, `NG_PPPOE_SESSION_NODE`, `PADI_CODE`, `pppoe_full_hdr::ph`, `sess_neg::pkt`, `packet::pkt_header`, `PPPOE_SINIT`, `PTT_HOST_UNIQ`, `sess_neg::service`, `sess_con::Session_ID`, and `sess_con::state`.

Referenced by `ng_pppoe_rcvmsg()`, and `pppoe_ticker()`.

Here is the call graph for this function:



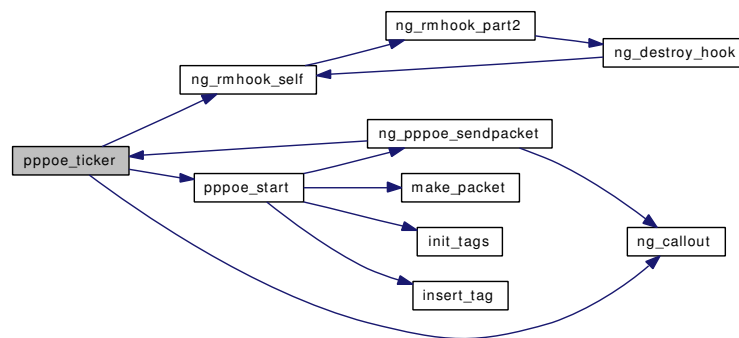
7.133.4.23 `static void pppoe_ticker (node_p node, hook_p hook, void * arg1, int arg2)` [static]

Definition at line 1781 of file `ng_pppoe.c`.

References `sess_neg::handle`, `ng_hook::hk_name`, `sess_neg::m`, `ng_node::nd_ID`, `sess_con::neg`, `ng_callout()`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `ng_rmhook_self()`, `NG_SEND_DATA_ONLY`, `PPPOE_PRIMED`, `PPPOE_SINIT`, `PPPOE_SOFFER`, `PPPOE_SREQ`, `pppoe_start()`, `PPPOE_TIMEOUT_LIMIT`, `sess_con::Session_ID`, `sess_con::state`, and `sess_neg::timeout`.

Referenced by `ng_pppoe_sendpacket()`.

Here is the call graph for this function:



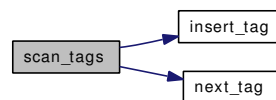
7.133.4.24 static struct `pppoe_tag` * `scan_tags` (`sessp sp`, const struct `pppoe_hdr` * `ph`) [static]

Definition at line 1890 of file `ng_pppoe.c`.

References `insert_tag()`, `next_tag()`, `PTT_AC_COOKIE`, `PTT_AC_NAME`, `PTT_EOL`, `PTT_GEN_ERR`, `PTT_HOST_UNIQ`, `PTT_RELAY_SID`, `PTT_SRV_ERR`, `PTT_SRV_NAME`, `PTT_SYS_ERR`, `PTT_VENDOR`, `sess_con::Session_ID`, `pppoe_hdr::tag`, `pppoe_tag::tag_len`, and `pppoe_tag::tag_type`.

Referenced by `ng_pppoe_rcvdata()`.

Here is the call graph for this function:



7.133.4.25 static int `send_acname` (`sessp sp`, const struct `pppoe_tag` * `tag`) [static]

Definition at line 1078 of file `ng_pppoe.c`.

References `sess_con::creator`, `ng_mesg::data`, `sess_con::hook`, `ngpppoe_sts::hook`, `min`, `NG_HOOK_NODE`, `NG_HOOKSIZ`, `NG_MKMESSAGE`, `NG_SEND_MSG_ID`, `NGM_PPPOE_ACNAME`, `NGM_PPPOE_COOKIE`, `sess_con::Session_ID`, `pppoe_tag::tag_data`, and `pppoe_tag::tag_len`.

Referenced by `ng_pppoe_rcvdata()`.

7.133.4.26 static int `send_sessionid` (`sessp sp`) [static]

Definition at line 1101 of file `ng_pppoe.c`.

References `sess_con::creator`, `ng_mesg::data`, `sess_con::hook`, `NG_HOOK_NODE`, `NG_MKMESSAGE`, `NG_SEND_MSG_ID`, `NGM_PPPOE_COOKIE`, `NGM_PPPOE_SESSIONID`, and `sess_con::Session_ID`.

Referenced by `ng_pppoe_rcvdata()`.

7.133.5 Variable Documentation

7.133.5.1 `struct ng_cmdlist ng_pppoe_cmds[]` [static]

Definition at line 99 of file ng_pppoe.c.

7.133.5.2 `ng_connect_t ng_pppoe_connect` [static]

Definition at line 78 of file ng_pppoe.c.

7.133.5.3 `ng_constructor_t ng_pppoe_constructor` [static]

Definition at line 74 of file ng_pppoe.c.

7.133.5.4 `ng_disconnect_t ng_pppoe_disconnect` [static]

Definition at line 80 of file ng_pppoe.c.

7.133.5.5 `ng_newhook_t ng_pppoe_newhook` [static]

Definition at line 77 of file ng_pppoe.c.

7.133.5.6 `ng_rcvdata_t ng_pppoe_rcvdata` [static]

Definition at line 79 of file ng_pppoe.c.

7.133.5.7 `ng_rcvmsg_t ng_pppoe_rcvmsg` [static]

Definition at line 75 of file ng_pppoe.c.

7.133.5.8 `ng_shutdown_t ng_pppoe_shutdown` [static]

Definition at line 76 of file ng_pppoe.c.

7.133.5.9 `struct ng_parse_type ng_pppoe_sts_state_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_pppoe_sts_type_fields
}
```

Definition at line 93 of file ng_pppoe.c.

7.133.5.10 `struct ng_parse_struct_field ng_pppoe_sts_type_fields[] = NG_PPPOE_STS_TYPE_INFO` [static]

Definition at line 92 of file ng_pppoe.c.

7.133.5.11 struct [ng_parse_type](#) [ngpppoe_init_data_state_type](#) [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ngpppoe_init_data_type_fields
}
```

Definition at line 85 of file ng_pppoe.c.

7.133.5.12 struct [ng_parse_struct_field](#) [ngpppoe_init_data_type_fields](#)[] = [NG_PPPOE_INIT_DATA_TYPE_INFO](#) [static]

Definition at line 84 of file ng_pppoe.c.

7.133.5.13 struct [ng_type](#) [typestruct](#) [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_PPPOE_NODE_TYPE,
    .constructor =  ng_pppoe_constructor,
    .rcvmsg =      ng_pppoe_rcvmsg,
    .shutdown =    ng_pppoe_shutdown,
    .newhook =     ng_pppoe_newhook,
    .connect =     ng_pppoe_connect,
    .rcvdata =     ng_pppoe_rcvdata,
    .disconnect =  ng_pppoe_disconnect,
    .cmdlist =     ng_pppoe_cmds,
}
```

Definition at line 174 of file ng_pppoe.c.

7.134 /usr/src/sys/netgraph/ng_pppoe.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngpppoestat](#)
- struct [ngpppoe_init_data](#)
- struct [ngpppoe_sts](#)
- struct [pppoe_tag](#)
- struct [pppoe_hdr](#)
- struct [pppoe_full_hdr](#)
- union [packet](#)
- struct [datatag](#)

Defines

- #define [NG_PPPOE_NODE_TYPE](#) "pppoe"
- #define [NGM_PPPOE_COOKIE](#) 1089893072
- #define [PPPOE_NUM_SESSIONS](#) 16
- #define [PPPOE_SERVICE_NAME_SIZE](#) 64
- #define [NG_PPPOE_HOOK_ETHERNET](#) "ethernet"
- #define [NG_PPPOE_HOOK_PADI](#) "PADI"
- #define [NG_PPPOE_HOOK_S_LEADIN](#) "service"
- #define [NG_PPPOE_HOOK_C_LEADIN](#) "client"
- #define [NG_PPPOE_HOOK_DEBUG](#) "debug"
- #define [NG_PPPOE_STANDARD](#) "standard"
- #define [NG_PPPOE_3COM](#) "3Com"
- #define [NG_PPPOE_NONSTANDARD](#) NG_PPPOE_3COM
- #define [NG_PPPOE_DLINK](#) "D-Link"
- #define [NG_PPPOESTAT_TYPE_INFO](#)
- #define [NG_PPPOE_INIT_DATA_TYPE_INFO](#)
- #define [NG_PPPOE_STS_TYPE_INFO](#)
- #define [PPPOE_TIMEOUT_LIMIT](#) 64
- #define [PPPOE_OFFER_TIMEOUT](#) 16
- #define [PPPOE_INITIAL_TIMEOUT](#) 2
- #define [PADI_CODE](#) 0x09
- #define [PADO_CODE](#) 0x07
- #define [PADR_CODE](#) 0x19
- #define [PADS_CODE](#) 0x65
- #define [PADT_CODE](#) 0xa7
- #define [PTT_EOL](#) (0x0000)
- #define [PTT_SRV_NAME](#) (0x0101)
- #define [PTT_AC_NAME](#) (0x0102)
- #define [PTT_HOST_UNIQ](#) (0x0103)
- #define [PTT_AC_COOKIE](#) (0x0104)

- #define `PTT_VENDOR` (0x0105)
- #define `PTT_RELAY_SID` (0x0110)
- #define `PTT_SRV_ERR` (0x0201)
- #define `PTT_SYS_ERR` (0x0202)
- #define `PTT_GEN_ERR` (0x0203)
- #define `ETHERTYPE_PPPOE_DISC` 0x8863
- #define `ETHERTYPE_PPPOE_SESS` 0x8864
- #define `ETHERTYPE_PPPOE_3COM_DISC` 0x3c12
- #define `ETHERTYPE_PPPOE_3COM_SESS` 0x3c13
- #define `TAGI_SVC` 0
- #define `TAGI_HUNIQ` 1
- #define `TAGO_ACNAME` 0
- #define `TAGO_SVC` 1
- #define `TAGO_COOKIE` 2
- #define `TAGO_HUNIQ` 3
- #define `TAGR_SVC` 0
- #define `TAGR_HUNIQ` 1
- #define `TAGR_COOKIE` 2
- #define `TAGS_ACNAME` 0
- #define `TAGS_SVC` 1
- #define `TAGS_COOKIE` 2
- #define `TAGS_HUNIQ` 3

Enumerations

- enum `cmd` {
 `NGM_PPPOE_SET_FLAG` = 1, `NGM_PPPOE_CONNECT` = 2, `NGM_PPPOE_LISTEN` = 3,
 `NGM_PPPOE_OFFER` = 4,
 `NGM_PPPOE_SUCCESS` = 5, `NGM_PPPOE_FAIL` = 6, `NGM_PPPOE_CLOSE` = 7, `NGM_PPPOE_SERVICE` = 8,
 `NGM_PPPOE_ACNAME` = 9, `NGM_PPPOE_GET_STATUS` = 10, `NGM_PPPOE_SESSIONID` = 11,
 `NGM_PPPOE_SETMODE` = 12,
 `NGM_PPPOE_GETMODE` = 13, `NGM_PPPOE_SETENADDR` = 14 }

Variables

- `pppoe_tag` __packed
- `pppoe_hdr` __packed
- `pppoe_full_hdr` __packed

7.134.1 Define Documentation

7.134.1.1 #define `ETHERTYPE_PPPOE_3COM_DISC` 0x3c12

Definition at line 186 of file `ng_pppoe.h`.

Referenced by `ng_pppoe_disconnect()`, `ng_pppoe_rcvdata()`, and `ng_pppoe_rcvmsg()`.

7.134.1.2 #define ETHERTYPE_PPPOE_3COM_SESS 0x3c13

Definition at line 187 of file ng_pppoe.h.

Referenced by ng_pppoe_disconnect(), and ng_pppoe_rcvdata().

7.134.1.3 #define ETHERTYPE_PPPOE_DISC 0x8863

Definition at line 184 of file ng_pppoe.h.

Referenced by ng_pppoe_constructor(), ng_pppoe_disconnect(), ng_pppoe_rcvdata(), and ng_pppoe_rcvmsg().

7.134.1.4 #define ETHERTYPE_PPPOE_SESS 0x8864

Definition at line 185 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata().

7.134.1.5 #define NG_PPPOE_3COM "3Com"

Definition at line 68 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvmsg().

7.134.1.6 #define NG_PPPOE_DLINK "D-Link"

Definition at line 70 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvmsg().

7.134.1.7 #define NG_PPPOE_HOOK_C_LEADIN "client"

Definition at line 63 of file ng_pppoe.h.

7.134.1.8 #define NG_PPPOE_HOOK_DEBUG "debug"

Definition at line 64 of file ng_pppoe.h.

Referenced by ng_pppoe_newhook().

7.134.1.9 #define NG_PPPOE_HOOK_ETHERNET "ethernet"

Definition at line 60 of file ng_pppoe.h.

Referenced by ng_pppoe_newhook().

7.134.1.10 #define NG_PPPOE_HOOK_PADI "PADI"

Definition at line 61 of file ng_pppoe.h.

7.134.1.11 #define NG_PPPOE_HOOK_S_LEADIN "service"

Definition at line 62 of file ng_pppoe.h.

7.134.1.12 #define NG_PPPOE_INIT_DATA_TYPE_INFO**Value:**

```
{
    \
    { "hook",      &ng_parse_hookbuf_type }, \
    { "data",     &ng_parse_sizedstring_type }, \
    { NULL }
}
```

Definition at line 134 of file ng_pppoe.h.

7.134.1.13 #define NG_PPPOE_NODE_TYPE "pppoe"

Definition at line 51 of file ng_pppoe.h.

7.134.1.14 #define NG_PPPOE_NONSTANDARD NG_PPPOE_3COM

Definition at line 69 of file ng_pppoe.h.

7.134.1.15 #define NG_PPPOE_STANDARD "standard"

Definition at line 67 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvmsg().

7.134.1.16 #define NG_PPPOE_STTS_TYPE_INFO**Value:**

```
{
    \
    { "hook",      &ng_parse_hookbuf_type }, \
    { NULL }
}
```

Definition at line 150 of file ng_pppoe.h.

7.134.1.17 #define NG_PPPOESTAT_TYPE_INFO**Value:**

```
{
    \
    { "packets_in", &ng_parse_uint_type }, \
    { "packets_out", &ng_parse_uint_type }, \
    { NULL }
}
```

Definition at line 103 of file ng_pppoe.h.

7.134.1.18 #define NGM_PPPOE_COOKIE 1089893072

Definition at line 53 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvmsg(), pppoe_send_event(), send_acname(), and send_sessionid().

7.134.1.19 #define PADI_CODE 0x09

Definition at line 165 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), and pppoe_start().

7.134.1.20 #define PADO_CODE 0x07

Definition at line 166 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), and ng_pppoe_rcvmsg().

7.134.1.21 #define PADR_CODE 0x19

Definition at line 167 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata().

7.134.1.22 #define PADS_CODE 0x65

Definition at line 168 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata().

7.134.1.23 #define PADT_CODE 0xa7

Definition at line 169 of file ng_pppoe.h.

Referenced by ng_pppoe_disconnect(), ng_pppoe_rcvdata(), and ng_pppoe_rcvmsg().

7.134.1.24 #define PPPOE_INITIAL_TIMEOUT 2

Definition at line 162 of file ng_pppoe.h.

Referenced by ng_pppoe_sendpacket().

7.134.1.25 #define PPPOE_NUM_SESSIONS 16

Definition at line 56 of file ng_pppoe.h.

7.134.1.26 #define PPPOE_OFFER_TIMEOUT 16

Definition at line 161 of file ng_pppoe.h.

Referenced by ng_pppoe_sendpacket().

7.134.1.27 #define PPPOE_SERVICE_NAME_SIZE 64

Definition at line 57 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvmsg().

7.134.1.28 #define PPPOE_TIMEOUT_LIMIT 64

Definition at line 160 of file ng_pppoe.h.

Referenced by pppoe_ticker().

7.134.1.29 #define PTT_AC_COOKIE (0x0104)

Definition at line 177 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), and scan_tags().

7.134.1.30 #define PTT_AC_NAME (0x0102)

Definition at line 175 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), and scan_tags().

7.134.1.31 #define PTT_EOL (0x0000)

Definition at line 173 of file ng_pppoe.h.

Referenced by scan_tags().

7.134.1.32 #define PTT_GEN_ERR (0x0203)

Definition at line 182 of file ng_pppoe.h.

Referenced by ng_pppoe_disconnect(), and scan_tags().

7.134.1.33 #define PTT_HOST_UNIQ (0x0103)

Definition at line 176 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), pppoe_start(), and scan_tags().

7.134.1.34 #define PTT_RELAY_SID (0x0110)

Definition at line 179 of file ng_pppoe.h.

Referenced by scan_tags().

7.134.1.35 #define PTT_SRV_ERR (0x0201)

Definition at line 180 of file ng_pppoe.h.

Referenced by scan_tags().

7.134.1.36 #define PTT_SRV_NAME (0x0101)

Definition at line 174 of file ng_pppoe.h.

Referenced by ng_pppoe_rcvdata(), ng_pppoe_rcvmsg(), and scan_tags().

7.134.1.37 #define PTT_SYS_ERR (0x0202)

Definition at line 181 of file ng_pppoe.h.

Referenced by scan_tags().

7.134.1.38 #define PTT_VENDOR (0x0105)

Definition at line 178 of file ng_pppoe.h.

Referenced by scan_tags().

7.134.1.39 #define TAGI_HUNIQ 1

Definition at line 244 of file ng_pppoe.h.

7.134.1.40 #define TAGI_SVC 0

Definition at line 243 of file ng_pppoe.h.

7.134.1.41 #define TAGO_ACNAME 0

Definition at line 246 of file ng_pppoe.h.

7.134.1.42 #define TAGO_COOKIE 2

Definition at line 248 of file ng_pppoe.h.

7.134.1.43 #define TAGO_HUNIQ 3

Definition at line 249 of file ng_pppoe.h.

7.134.1.44 #define TAGO_SVC 1

Definition at line 247 of file ng_pppoe.h.

7.134.1.45 #define TAGR_COOKIE 2

Definition at line 253 of file ng_pppoe.h.

7.134.1.46 #define TAGR_HUNIQ 1

Definition at line 252 of file ng_pppoe.h.

7.134.1.47 #define TAGR_SVC 0

Definition at line 251 of file ng_pppoe.h.

7.134.1.48 #define TAGS_ACNAME 0

Definition at line 255 of file ng_pppoe.h.

7.134.1.49 #define TAGS_COOKIE 2

Definition at line 257 of file ng_pppoe.h.

7.134.1.50 #define TAGS_HUNIQ 3

Definition at line 258 of file ng_pppoe.h.

7.134.1.51 #define TAGS_SVC 1

Definition at line 256 of file ng_pppoe.h.

7.134.2 Enumeration Type Documentation**7.134.2.1 enum [cmd](#)**

Enumerator:

NGM_PPPOE_SET_FLAG
NGM_PPPOE_CONNECT
NGM_PPPOE_LISTEN
NGM_PPPOE_OFFER
NGM_PPPOE_SUCCESS
NGM_PPPOE_FAIL
NGM_PPPOE_CLOSE
NGM_PPPOE_SERVICE
NGM_PPPOE_ACNAME
NGM_PPPOE_GET_STATUS
NGM_PPPOE_SESSIONID
NGM_PPPOE_SETMODE
NGM_PPPOE_GETMODE
NGM_PPPOE_SETENADDR

Definition at line 76 of file ng_pppoe.h.

7.134.3 Variable Documentation

7.134.3.1 struct [pppoe_full_hdr __packed](#)

7.134.3.2 struct [pppoe_hdr __packed](#)

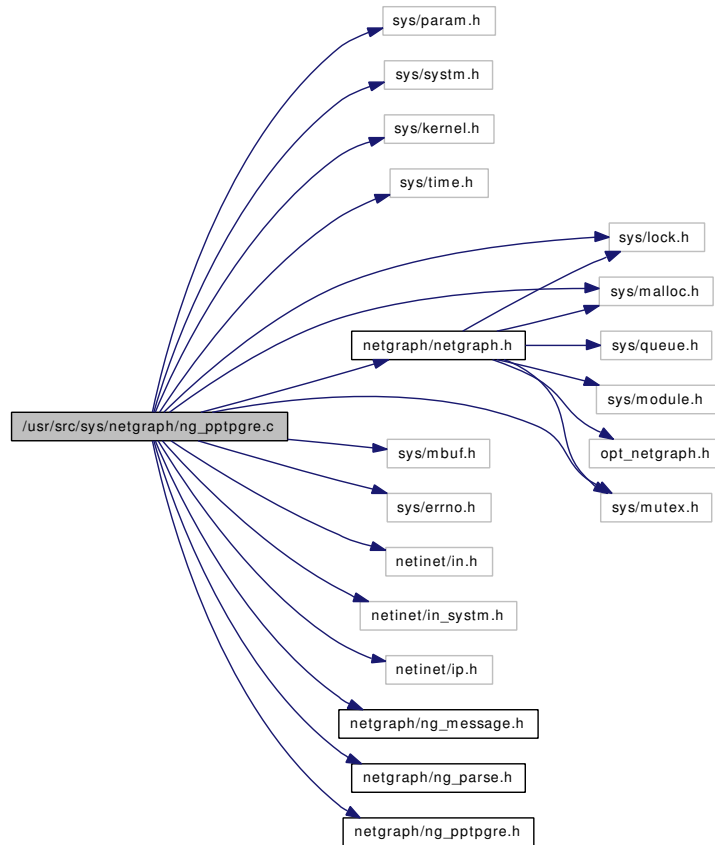
7.134.3.3 struct [pppoe_tag __packed](#)

Referenced by `ng_pppoe_rcvdata()`, and `pppoe_start()`.

7.135 /usr/src/sys/netgraph/ng_pptpgre.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/time.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/errno.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_pptpgre.h>
```

Include dependency graph for ng_pptpgre.c:



Data Structures

- struct [greheader](#)
- struct [ng_pptpgre_ackp](#)
- struct [ng_pptpgre_private](#)

Defines

- #define [PPTP_GRE_PROTO](#) 0x880b
- #define [PPTP_INIT_VALUE](#) ((0x2001 << 16) | PPTP_GRE_PROTO)
- #define [PPTP_INIT_MASK](#) 0xef7ffff
- #define [PPTP_MAX_PAYLOAD](#) (0xffff - sizeof(struct greheader) - 8)
- #define [PPTP_TIME_SCALE](#) 1000
- #define [PPTP_XMIT_WIN](#) 16
- #define [PPTP_MIN_RTT](#) (PPTP_TIME_SCALE / 10)
- #define [PPTP_MIN_TIMEOUT](#) (PPTP_TIME_SCALE / 83)
- #define [PPTP_MAX_TIMEOUT](#) (3 * PPTP_TIME_SCALE)
- #define [PPTP_MIN_ACK_DELAY](#) (PPTP_TIME_SCALE / 500)
- #define [PPTP_MAX_ACK_DELAY](#) (PPTP_TIME_SCALE / 2)
- #define [PPTP_ACK_ALPHA](#)(x) (((x) + 4) >> 3)
- #define [PPTP_ACK_BETA](#)(x) (((x) + 2) >> 2)
- #define [PPTP_ACK_CHI](#)(x) ((x) << 2)

- #define [PPTP_ACK_DELTA](#)(x) ((x) << 1)
- #define [PPTP_SEQ_DIFF](#)(x, y) ((int32_t)(x) - (int32_t)(y))
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef u_int64_t [pptptime_t](#)
- typedef [ng_pptpgre_private](#) * [priv_p](#)

Functions

- static int [ng_pptpgre_xmit](#) ([node_p](#) node, [item_p](#) item)
- static int [ng_pptpgre_rcv](#) ([node_p](#) node, [item_p](#) item)
- static void [ng_pptpgre_start_send_ack_timer](#) ([node_p](#) node, int ackTimeout)
- static void [ng_pptpgre_stop_send_ack_timer](#) ([node_p](#) node)
- static void [ng_pptpgre_start_rcv_ack_timer](#) ([node_p](#) node)
- static void [ng_pptpgre_stop_rcv_ack_timer](#) ([node_p](#) node)
- static void [ng_pptpgre_rcv_ack_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static void [ng_pptpgre_send_ack_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- static void [ng_pptpgre_reset](#) ([node_p](#) node)
- static [pptptime_t](#) [ng_pptpgre_time](#) ([node_p](#) node)
- [NETGRAPH_INIT](#) (pptpgre,&[ng_pptpgre_tpestruct](#))
- static int [ng_pptpgre_constructor](#) ([node_p](#) node)
- static int [ng_pptpgre_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_pptpgre_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_pptpgre_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_pptpgre_shutdown](#) ([node_p](#) node)
- static int [ng_pptpgre_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_pptpgre_constructor](#)
- static [ng_rcvmsg_t](#) [ng_pptpgre_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_pptpgre_shutdown](#)
- static [ng_newhook_t](#) [ng_pptpgre_newhook](#)
- static [ng_rcvdata_t](#) [ng_pptpgre_rcvdata](#)
- static [ng_disconnect_t](#) [ng_pptpgre_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_pptpgre_conf_type_fields](#) [] = [NG_PPTPGRE_CONF_TYPE_INFO](#)
- static struct [ng_parse_type](#) [ng_pptpgre_conf_type](#)
- static struct [ng_parse_struct_field](#) [ng_pptpgre_stats_type_fields](#) [] = [NG_PPTPGRE_STATS_TYPE_INFO](#)
- static struct [ng_parse_type](#) [ng_pptpgre_stats_type](#)
- static struct [ng_cmdlist](#) [ng_pptpgre_cmdlist](#) []
- static struct [ng_type](#) [ng_pptpgre_tpestruct](#)

7.135.1 Define Documentation

7.135.1.1 **#define ERROUT(x) do { error = (x); goto done; } while (0)**

Definition at line 266 of file ng_pptpgre.c.

7.135.1.2 **#define PPTP_ACK_ALPHA(x) (((x) + 4) >> 3)**

Definition at line 135 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv().

7.135.1.3 **#define PPTP_ACK_BETA(x) (((x) + 2) >> 2)**

Definition at line 136 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv().

7.135.1.4 **#define PPTP_ACK_CHI(x) ((x) << 2)**

Definition at line 137 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_rcv_ack_timeout().

7.135.1.5 **#define PPTP_ACK_DELTA(x) ((x) << 1)**

Definition at line 138 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv_ack_timeout().

7.135.1.6 **#define PPTP_GRE_PROTO 0x880b**

Definition at line 108 of file ng_pptpgre.c.

7.135.1.7 **#define PPTP_INIT_MASK 0xef7ffff**

Definition at line 112 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv().

7.135.1.8 **#define PPTP_INIT_VALUE ((0x2001 << 16) | PPTP_GRE_PROTO)**

Definition at line 111 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_xmit().

7.135.1.9 **#define PPTP_MAX_ACK_DELAY (PPTP_TIME_SCALE / 2)**

Definition at line 132 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv().

7.135.1.10 #define PPTP_MAX_PAYLOAD (0xffff - sizeof(struct greheader) - 8)

Definition at line 115 of file ng_pptpgre.c.

Referenced by ng_pptpgre_xmit().

7.135.1.11 #define PPTP_MAX_TIMEOUT (3 * PPTP_TIME_SCALE)

Definition at line 125 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), ng_pptpgre_rcv_ack_timeout(), and ng_pptpgre_reset().

7.135.1.12 #define PPTP_MIN_ACK_DELAY (PPTP_TIME_SCALE / 500)

Definition at line 131 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv().

7.135.1.13 #define PPTP_MIN_RTT (PPTP_TIME_SCALE / 10)

Definition at line 123 of file ng_pptpgre.c.

Referenced by ng_pptpgre_reset().

7.135.1.14 #define PPTP_MIN_TIMEOUT (PPTP_TIME_SCALE / 83)

Definition at line 124 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_rcv_ack_timeout().

7.135.1.15 #define PPTP_SEQ_DIFF(x, y) ((int32_t)(x) - (int32_t)(y))

Definition at line 140 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_xmit().

7.135.1.16 #define PPTP_TIME_SCALE 1000

Definition at line 118 of file ng_pptpgre.c.

Referenced by ng_pptpgre_reset(), ng_pptpgre_start_rcv_ack_timer(), ng_pptpgre_start_send_ack_timer(), and ng_pptpgre_time().

7.135.1.17 #define PPTP_XMIT_WIN 16

Definition at line 122 of file ng_pptpgre.c.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_reset().

7.135.2 Typedef Documentation

7.135.2.1 typedef u_int64_t ptpptime_t

Definition at line 119 of file ng_ptpgre.c.

7.135.2.2 typedef struct ng_ptpgre_private* priv_p

Definition at line 172 of file ng_ptpgre.c.

7.135.3 Function Documentation

7.135.3.1 NETGRAPH_INIT (ptpgre, & ng_ptpgre_tpestruct)

7.135.3.2 static int ng_ptpgre_constructor (node_p node) [static]

Definition at line 276 of file ng_ptpgre.c.

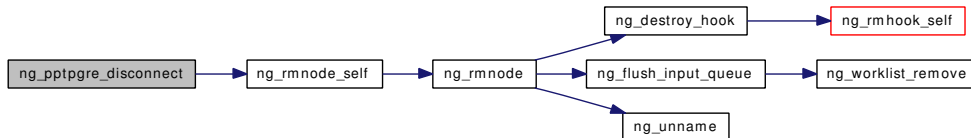
References ng_callout_init, and NG_NODE_SET_PRIVATE.

7.135.3.3 static int ng_ptpgre_disconnect (hook_p hook) [static]

Definition at line 441 of file ng_ptpgre.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.135.3.4 static int ng_ptpgre_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 300 of file ng_ptpgre.c.

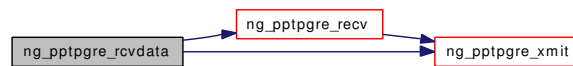
References NG_NODE_PRIVATE, NG_PPTPGRE_HOOK_LOWER, and NG_PPTPGRE_HOOK_UPPER.

7.135.3.5 static int ng_ptpgre_rcvdata (hook_p hook, item_p item) [static]

Definition at line 390 of file ng_ptpgre.c.

References NG_FREE_ITEM, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_ptpgre_rcv(), and ng_ptpgre_xmit().

Here is the call graph for this function:

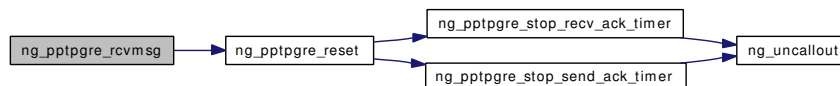


7.135.3.6 static int ng_pptpgre_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 326 of file ng_pptpgre.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERROUT, ng_mesg::header, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, ng_pptpgre_reset(), NG_RESPOND_MSG, NGI_GET_MSG, NGM_PPTPGRE_CLR_STATS, NGM_PPTPGRE_COOKIE, NGM_PPTPGRE_GET_CONFIG, NGM_PPTPGRE_GET_STATS, NGM_PPTPGRE_GETCLR_STATS, NGM_PPTPGRE_SET_CONFIG, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



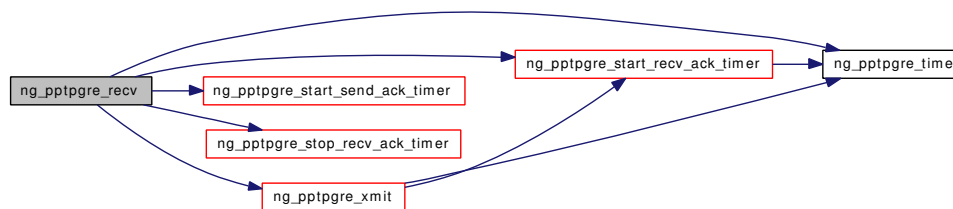
7.135.3.7 static int ng_pptpgre_rcv (node_p node, item_p item) [static]

Definition at line 585 of file ng_pptpgre.c.

References ng_pptpgre_ackp::ato, ng_pptpgre_ackp::dev, ERROUT, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_NODE_PRIVATE, ng_pptpgre_start_rcv_ack_timer(), ng_pptpgre_start_send_ack_timer(), ng_pptpgre_stop_rcv_ack_timer(), ng_pptpgre_time(), ng_pptpgre_xmit(), NGI_GET_M, PPTP_ACK_ALPHA, PPTP_ACK_BETA, PPTP_ACK_CHI, PPTP_INIT_MASK, PPTP_INIT_VALUE, PPTP_MAX_ACK_DELAY, PPTP_MAX_TIMEOUT, PPTP_MIN_ACK_DELAY, PPTP_MIN_TIMEOUT, PPTP_SEQ_DIFF, PPTP_XMIT_WIN, ng_pptpgre_ackp::rtt, ng_pptpgre_ackp::sackTimer, ng_pptpgre_ackp::timeSent, ng_pptpgre_ackp::winAck, and ng_pptpgre_ackp::xmitWin.

Referenced by ng_pptpgre_rcvdata().

Here is the call graph for this function:



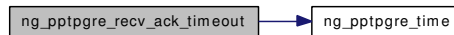
7.135.3.8 `static void ng_pptpgre_rcv_ack_timeout (node_p node, hook_p hook, void * arg1, int arg2) [static]`

Definition at line 822 of file ng_pptpgre.c.

References `ng_pptpgre_ackp::ato`, `ng_pptpgre_ackp::dev`, `NG_NODE_PRIVATE`, `ng_pptpgre_time()`, `PPTP_ACK_CHI`, `PPTP_ACK_DELTA`, `PPTP_MAX_TIMEOUT`, `PPTP_MIN_TIMEOUT`, `ng_pptpgre_ackp::rtt`, `ng_pptpgre_ackp::timeSent`, `ng_pptpgre_ackp::winAck`, and `ng_pptpgre_ackp::xmitWin`.

Referenced by `ng_pptpgre_start_rcv_ack_timer()`.

Here is the call graph for this function:



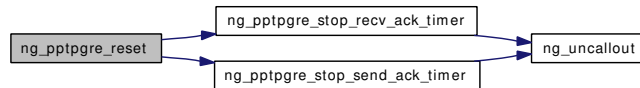
7.135.3.9 `static void ng_pptpgre_reset (node_p node) [static]`

Definition at line 907 of file ng_pptpgre.c.

References `ng_pptpgre_ackp::ato`, `ng_pptpgre_ackp::dev`, `NG_NODE_PRIVATE`, `ng_pptpgre_stop_rcv_ack_timer()`, `ng_pptpgre_stop_send_ack_timer()`, `PPTP_MAX_TIMEOUT`, `PPTP_MIN_RTT`, `PPTP_TIME_SCALE`, `PPTP_XMIT_WIN`, `ng_pptpgre_ackp::rtt`, `ng_pptpgre_ackp::winAck`, and `ng_pptpgre_ackp::xmitWin`.

Referenced by `ng_pptpgre_rcvmsg()`, and `ng_pptpgre_shutdown()`.

Here is the call graph for this function:



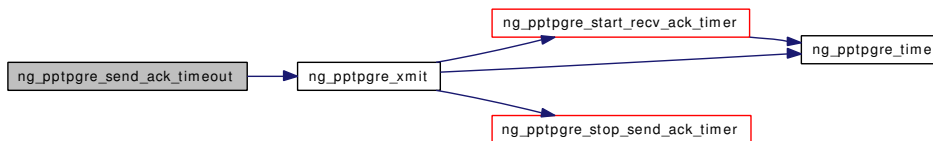
7.135.3.10 `static void ng_pptpgre_send_ack_timeout (node_p node, hook_p hook, void * arg1, int arg2) [static]`

Definition at line 889 of file ng_pptpgre.c.

References `NG_NODE_PRIVATE`, and `ng_pptpgre_xmit()`.

Referenced by `ng_pptpgre_start_send_ack_timer()`.

Here is the call graph for this function:

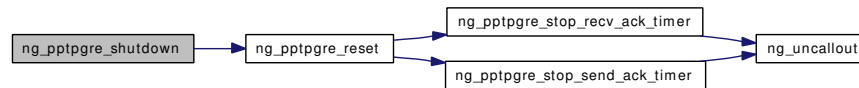


7.135.3.11 static int ng_pptpgre_shutdown (node_p node) [static]

Definition at line 421 of file ng_pptpgre.c.

References NG_NODE_PRIVATE, NG_NODE_UNREF, and ng_pptpgre_reset().

Here is the call graph for this function:

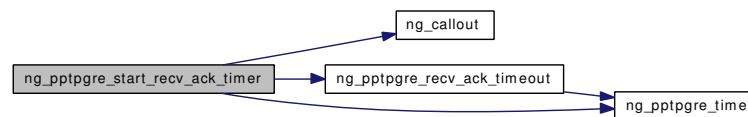
**7.135.3.12 static void ng_pptpgre_start_rcv_ack_timer (node_p node) [static]**

Definition at line 776 of file ng_pptpgre.c.

References ng_pptpgre_ackp::ato, ng_callout(), NG_NODE_PRIVATE, ng_pptpgre_rcv_ack_timeout(), ng_pptpgre_time(), PPTP_TIME_SCALE, ng_pptpgre_ackp::rackTimer, and ng_pptpgre_ackp::timeSent.

Referenced by ng_pptpgre_rcv(), and ng_pptpgre_xmit().

Here is the call graph for this function:

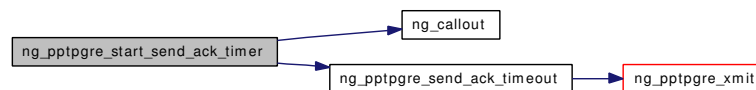
**7.135.3.13 static void ng_pptpgre_start_send_ack_timer (node_p node, int ackTimeout) [static]**

Definition at line 858 of file ng_pptpgre.c.

References ng_callout(), NG_NODE_PRIVATE, ng_pptpgre_send_ack_timeout(), PPTP_TIME_SCALE, and ng_pptpgre_ackp::sackTimer.

Referenced by ng_pptpgre_rcv().

Here is the call graph for this function:

**7.135.3.14 static void ng_pptpgre_stop_rcv_ack_timer (node_p node) [static]**

Definition at line 805 of file ng_pptpgre.c.

References NG_NODE_PRIVATE, ng_uncallout(), and ng_pptpgre_ackp::rackTimer.

Referenced by `ng_pptpgre_rcv()`, and `ng_pptpgre_reset()`.

Here is the call graph for this function:



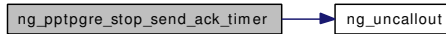
7.135.3.15 `static void ng_pptpgre_stop_send_ack_timer (node_p node) [static]`

Definition at line 874 of file `ng_pptpgre.c`.

References `NG_NODE_PRIVATE`, `ng_unccallout()`, and `ng_pptpgre_ackp::sackTimer`.

Referenced by `ng_pptpgre_reset()`, and `ng_pptpgre_xmit()`.

Here is the call graph for this function:



7.135.3.16 `static pptptime_t ng_pptpgre_time (node_p node) [static]`

Definition at line 950 of file `ng_pptpgre.c`.

References `NG_NODE_PRIVATE`, and `PPTP_TIME_SCALE`.

Referenced by `ng_pptpgre_rcv()`, `ng_pptpgre_rcv_ack_timeout()`, `ng_pptpgre_start_rcv_ack_timer()`, and `ng_pptpgre_xmit()`.

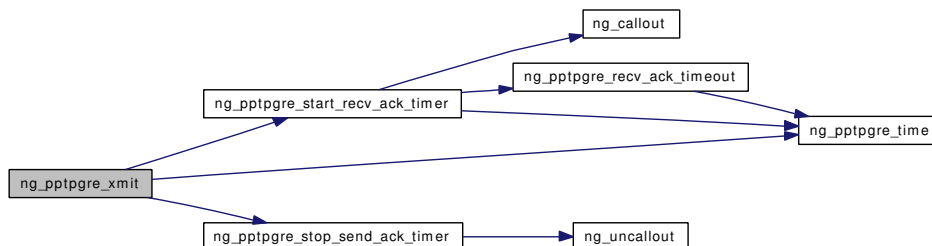
7.135.3.17 `static int ng_pptpgre_xmit (node_p node, item_p item) [static]`

Definition at line 469 of file `ng_pptpgre.c`.

References `greheader::cid`, `greheader::data`, `ERROUT`, `greheader::hasAck`, `greheader::hasSeq`, `greheader::length`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_NODE_PRIVATE`, `ng_pptpgre_start_rcv_ack_timer()`, `ng_pptpgre_stop_send_ack_timer()`, `ng_pptpgre_time()`, `NG_SEND_DATA_ONLY`, `NGI_GET_M`, `PPTP_INIT_VALUE`, `PPTP_MAX_PAYLOAD`, `PPTP_SEQ_DIFF`, `ng_pptpgre_ackp::timeSent`, and `ng_pptpgre_ackp::xmitWin`.

Referenced by `ng_pptpgre_rcvdata()`, `ng_pptpgre_rcv()`, and `ng_pptpgre_send_ack_timeout()`.

Here is the call graph for this function:



7.135.4 Variable Documentation

7.135.4.1 struct [ng_parse_type](#) [ng_pptp_stats_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_pptpgre_stats_type_fields
}
```

Definition at line 207 of file ng_pptpgre.c.

7.135.4.2 struct [ng_cmdlist](#) [ng_pptpgre_cmdlist](#)[] [static]

Definition at line 213 of file ng_pptpgre.c.

7.135.4.3 struct [ng_parse_type](#) [ng_pptpgre_conf_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_pptpgre_conf_type_fields,
}
```

Definition at line 199 of file ng_pptpgre.c.

7.135.4.4 struct [ng_parse_struct_field](#) [ng_pptpgre_conf_type_fields](#)[] = [NG_PPTPGRE_CONF_TYPE_INFO](#) [static]

Definition at line 198 of file ng_pptpgre.c.

7.135.4.5 [ng_constructor_t](#) [ng_pptpgre_constructor](#) [static]

Definition at line 175 of file ng_pptpgre.c.

7.135.4.6 [ng_disconnect_t](#) [ng_pptpgre_disconnect](#) [static]

Definition at line 180 of file ng_pptpgre.c.

7.135.4.7 [ng_newhook_t](#) [ng_pptpgre_newhook](#) [static]

Definition at line 178 of file ng_pptpgre.c.

7.135.4.8 [ng_rcvdata_t](#) [ng_pptpgre_rcvdata](#) [static]

Definition at line 179 of file ng_pptpgre.c.

7.135.4.9 `ng_rcvmsg_t ng_pptpgre_rcvmsg` [static]

Definition at line 176 of file `ng_pptpgre.c`.

7.135.4.10 `ng_shutdown_t ng_pptpgre_shutdown` [static]

Definition at line 177 of file `ng_pptpgre.c`.

7.135.4.11 `struct ng_parse_struct_field ng_pptpgre_stats_type_fields[] = NG_PPTPGRE_STATS_TYPE_INFO` [static]

Definition at line 206 of file `ng_pptpgre.c`.

7.135.4.12 `struct ng_type ng_pptpgre_typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_PPTPGRE_NODE_TYPE,
    .constructor = ng_pptpgre_constructor,
    .rcvmsg =      ng_pptpgre_rcvmsg,
    .shutdown =    ng_pptpgre_shutdown,
    .newhook =     ng_pptpgre_newhook,
    .rcvdata =     ng_pptpgre_rcvdata,
    .disconnect =  ng_pptpgre_disconnect,
    .cmdlist =     ng_pptpgre_cmdlist,
}
```

Definition at line 253 of file `ng_pptpgre.c`.

7.136 /usr/src/sys/netgraph/ng_pptpgre.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_pptpgre_conf](#)
- struct [ng_pptpgre_stats](#)

Defines

- #define [NG_PPTPGRE_NODE_TYPE](#) "pptpgre"
- #define [NGM_PPTPGRE_COOKIE](#) 1082548365
- #define [NG_PPTPGRE_HOOK_UPPER](#) "upper"
- #define [NG_PPTPGRE_HOOK_LOWER](#) "lower"
- #define [NG_PPTPGRE_CONF_TYPE_INFO](#)
- #define [NG_PPTPGRE_STATS_TYPE_INFO](#)

Enumerations

- enum {
 - [NGM_PPTPGRE_SET_CONFIG](#) = 1, [NGM_PPTPGRE_GET_CONFIG](#), [NGM_PPTPGRE_GET_STATS](#), [NGM_PPTPGRE_CLR_STATS](#),
 - [NGM_PPTPGRE_GETCLR_STATS](#) }

7.136.1 Define Documentation

7.136.1.1 #define NG_PPTPGRE_CONF_TYPE_INFO

Value:

```

{
    { "enabled",          &ng_parse_uint8_type }, \
    { "enableDelayedAck", &ng_parse_uint8_type }, \
    { "enableAlwaysAck", &ng_parse_uint8_type }, \
    { "enableWindowing", &ng_parse_uint8_type }, \
    { "cid",             &ng_parse_hint16_type }, \
    { "peerCid",         &ng_parse_hint16_type }, \
    { "recvWin",         &ng_parse_uint16_type }, \
    { "peerPpd",         &ng_parse_uint16_type }, \
    { NULL }
}
  
```

Definition at line 69 of file `ng_pptpgre.h`.

7.136.1.2 #define NG_PPTPGRE_HOOK_LOWER "lower"

Definition at line 53 of file ng_pptpgre.h.

Referenced by ng_pptpgre_newhook().

7.136.1.3 #define NG_PPTPGRE_HOOK_UPPER "upper"

Definition at line 52 of file ng_pptpgre.h.

Referenced by ng_pptpgre_newhook().

7.136.1.4 #define NG_PPTPGRE_NODE_TYPE "pptpgre"

Definition at line 48 of file ng_pptpgre.h.

7.136.1.5 #define NG_PPTPGRE_STATS_TYPE_INFO

Value:

```
{
    { "xmitPackets",      &ng_parse_uint32_type }, \
    { "xmitOctets",      &ng_parse_uint32_type }, \
    { "xmitLoneAcks",    &ng_parse_uint32_type }, \
    { "xmitDrops",       &ng_parse_uint32_type }, \
    { "xmitTooBig",      &ng_parse_uint32_type }, \
    { "recvPackets",     &ng_parse_uint32_type }, \
    { "recvOctets",      &ng_parse_uint32_type }, \
    { "recvRunts",       &ng_parse_uint32_type }, \
    { "recvBadGRE",      &ng_parse_uint32_type }, \
    { "recvBadAcks",     &ng_parse_uint32_type }, \
    { "recvBadCID",      &ng_parse_uint32_type }, \
    { "recvOutOfOrder", &ng_parse_uint32_type }, \
    { "recvDuplicates", &ng_parse_uint32_type }, \
    { "recvLoneAcks",    &ng_parse_uint32_type }, \
    { "recvAckTimeouts", &ng_parse_uint32_type }, \
    { "memoryFailures", &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 102 of file ng_pptpgre.h.

7.136.1.6 #define NGM_PPTPGRE_COOKIE 1082548365

Definition at line 49 of file ng_pptpgre.h.

Referenced by ng_pptpgre_rcvmsg().

7.136.2 Enumeration Type Documentation**7.136.2.1 anonymous enum**

Enumerator:

NGM_PPTPGRE_SET_CONFIG

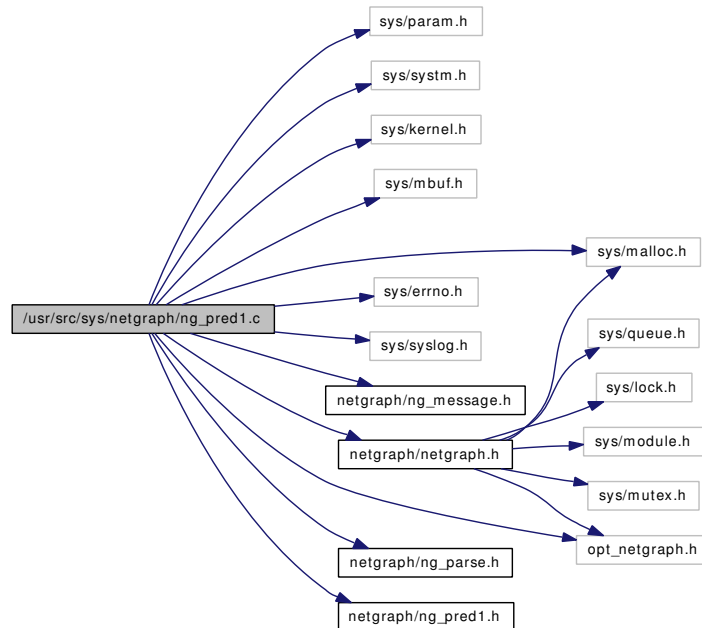
NGM_PPTPGRE_GET_CONFIG
NGM_PPTPGRE_GET_STATS
NGM_PPTPGRE_CLR_STATS
NGM_PPTPGRE_GETCLR_STATS

Definition at line 123 of file ng_pptpgre.h.

7.137 /usr/src/sys/netgraph/ng_pred1.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_pred1.h>
#include "opt_netgraph.h"
```

Include dependency graph for ng_pred1.c:



Data Structures

- struct [ng_pred1_private](#)

Defines

- #define [PRED1_HDRLEN](#) 2
- #define [PRED1_TABLE_SIZE](#) 0x10000

- #define `PRED1_BUF_SIZE` 4096
- #define `PPP_INITFCS` 0xffff
- #define `PPP_GOODFCS` 0xf0b8
- #define `HASH(x) priv` → Hash = (priv → Hash << 4) ^ (x)
- #define `ERROUT(x) do { error = (x); goto done; } while (0)`

Typedefs

- typedef `ng_pred1_private * priv_p`

Functions

- `MALLOC_DEFINE` (M_NETGRAPH_PRED1,"netgraph_pred1","netgraph pred1 node ")
- static int `ng_pred1_compress` (`node_p` node, struct mbuf *m, struct mbuf **resultp)
- static int `ng_pred1_decompress` (`node_p` node, struct mbuf *m, struct mbuf **resultp)
- static void `Pred1Init` (`node_p` node)
- static int `Pred1Compress` (`node_p` node, u_char *source, u_char *dest, int len)
- static int `Pred1Decompress` (`node_p` node, u_char *source, u_char *dest, int slen, int dlen)
- static void `Pred1SyncTable` (`node_p` node, u_char *source, int len)
- static uint16_t `Crc16` (uint16_t fcs, u_char *cp, int len)
- `NETGRAPH_INIT` (pred1,&ng_pred1_typestruct)
- static int `ng_pred1_constructor` (`node_p` node)
- static int `ng_pred1_newhook` (`node_p` node, `hook_p` hook, const char *name)
- static int `ng_pred1_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_pred1_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_pred1_shutdown` (`node_p` node)
- static int `ng_pred1_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_pred1_constructor`
- static `ng_rcvmsg_t` `ng_pred1_rcvmsg`
- static `ng_shutdown_t` `ng_pred1_shutdown`
- static `ng_newhook_t` `ng_pred1_newhook`
- static `ng_rcvdata_t` `ng_pred1_rcvdata`
- static `ng_disconnect_t` `ng_pred1_disconnect`
- static const uint16_t `Crc16Table` []
- static struct `ng_parse_struct_field` `ng_pred1_config_type_fields` [] = NG_PRED1_CONFIG_INFO
- static struct `ng_parse_type` `ng_pred1_config_type`
- static struct `ng_parse_struct_field` `ng_pred1_stats_type_fields` [] = NG_PRED1_STATS_INFO
- static struct `ng_parse_type` `ng_pred1_stat_type`
- static struct `ng_cmdlist` `ng_pred1_cmds` []
- static struct `ng_type` `ng_pred1_typestruct`

7.137.1 Define Documentation

7.137.1.1 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 176 of file `ng_pred1.c`.

7.137.1.2 #define HASH(x) *priv* → Hash = (*priv* → Hash << 4) ^ (x)

Definition at line 67 of file ng_pred1.c.

7.137.1.3 #define PPP_GOODFCS 0xf0b8

Definition at line 57 of file ng_pred1.c.

Referenced by ng_pred1_decompress(), and nga_rcv_async().

7.137.1.4 #define PPP_INITFCS 0xffff

Definition at line 56 of file ng_pred1.c.

Referenced by ng_pred1_compress(), ng_pred1_decompress(), and nga_rcv_sync().

7.137.1.5 #define PRED1_BUF_SIZE 4096

Definition at line 55 of file ng_pred1.c.

Referenced by ng_pred1_compress(), and ng_pred1_decompress().

7.137.1.6 #define PRED1_HDRLEN 2

Definition at line 52 of file ng_pred1.c.

7.137.1.7 #define PRED1_TABLE_SIZE 0x10000

Definition at line 54 of file ng_pred1.c.

Referenced by Pred1Init().

7.137.2 Typedef Documentation**7.137.2.1 typedef struct ng_pred1_private* *priv_p***

Definition at line 81 of file ng_pred1.c.

7.137.3 Function Documentation**7.137.3.1 static uint16_t Crc16 (uint16_t *fcs*, u_char * *cp*, int *len*) [static]**

Definition at line 657 of file ng_pred1.c.

References Crc16Table.

Referenced by ng_pred1_compress(), and ng_pred1_decompress().

7.137.3.2 `MALLOC_DEFINE (M_NETGRAPH_PRED1, "netgraph_pred1", "netgraph pred1 node ")`

7.137.3.3 `NETGRAPH_INIT (pred1, & ng_pred1_typestruct)`

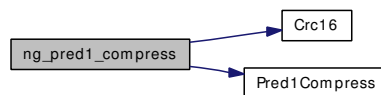
7.137.3.4 `static int ng_pred1_compress (node_p node, struct mbuf * m, struct mbuf ** resultp)`
[static]

Definition at line 380 of file ng_pred1.c.

References `Crc16()`, `NG_FREE_M`, `NG_NODE_PRIVATE`, `PPP_INITFCS`, `PRED1_BUF_SIZE`, and `Pred1Compress()`.

Referenced by `ng_pred1_rcvdata()`.

Here is the call graph for this function:



7.137.3.5 `static int ng_pred1_constructor (node_p node)` [static]

Definition at line 186 of file ng_pred1.c.

References `NG_NODE_FORCE_WRITER`, and `NG_NODE_SET_PRIVATE`.

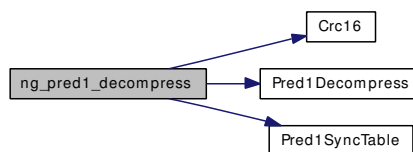
7.137.3.6 `static int ng_pred1_decompress (node_p node, struct mbuf * m, struct mbuf ** resultp)`
[static]

Definition at line 456 of file ng_pred1.c.

References `Crc16()`, `NG_FREE_M`, `NG_NODE_PRIVATE`, `PPP_GOODFCS`, `PPP_INITFCS`, `PRED1_BUF_SIZE`, `Pred1Decompress()`, and `Pred1SyncTable()`.

Referenced by `ng_pred1_rcvdata()`.

Here is the call graph for this function:

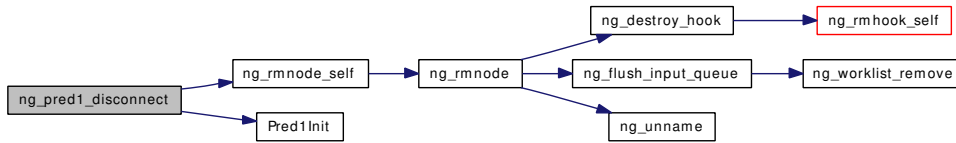


7.137.3.7 `static int ng_pred1_disconnect (hook_p hook)` [static]

Definition at line 359 of file ng_pred1.c.

References `NG_HOOK_NODE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `ng_rmnode_self()`, and `Pred1Init()`.

Here is the call graph for this function:



7.137.3.8 static int ng_pred1_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 206 of file ng_pred1.c.

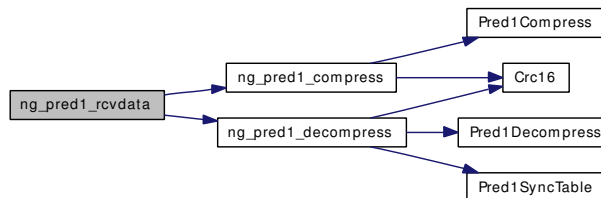
References NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, NG_PRED1_HOOK_COMP, and NG_PRED1_HOOK_DECOMP.

7.137.3.9 static int ng_pred1_rcvdata (hook_p hook, item_p item) [static]

Definition at line 297 of file ng_pred1.c.

References NG_FREE_ITEM, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_MKMESSAGE, NG_NODE_PRIVATE, ng_pred1_compress(), ng_pred1_decompress(), NG_SEND_MSG_ID, NGI_GET_M, NGM_PRED1_COOKIE, and NGM_PRED1_RESETREQ.

Here is the call graph for this function:

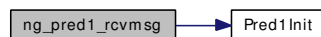


7.137.3.10 static int ng_pred1_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 227 of file ng_pred1.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERRROUT, ng_mesg::header, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGI_RETADDR, NGM_PRED1_CLR_STATS, NGM_PRED1_CONFIG, NGM_PRED1_COOKIE, NGM_PRED1_GET_STATS, NGM_PRED1_GETCLR_STATS, NGM_PRED1_RESETREQ, Pred1Init(), and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



7.137.3.11 `static int ng_pred1_shutdown (node_p node)` [static]

Definition at line 345 of file ng_pred1.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.137.3.12 `static int Pred1Compress (node_p node, u_char * source, u_char * dest, int len)`
[static]

Definition at line 565 of file ng_pred1.c.

References HASH, and NG_NODE_PRIVATE.

Referenced by ng_pred1_compress().

7.137.3.13 `static int Pred1Decompress (node_p node, u_char * source, u_char * dest, int slen, int dlen)` [static]

Definition at line 600 of file ng_pred1.c.

References HASH, and NG_NODE_PRIVATE.

Referenced by ng_pred1_decompress().

7.137.3.14 `static void Pred1Init (node_p node)` [static]

Definition at line 552 of file ng_pred1.c.

References NG_NODE_PRIVATE, and PRED1_TABLE_SIZE.

Referenced by ng_pred1_disconnect(), and ng_pred1_rcvmsg().

7.137.3.15 `static void Pred1SyncTable (node_p node, u_char * source, int len)` [static]

Definition at line 639 of file ng_pred1.c.

References HASH, and NG_NODE_PRIVATE.

Referenced by ng_pred1_decompress().

7.137.4 Variable Documentation**7.137.4.1** `static const uint16_t Crc16Table` [static]

Definition at line 104 of file ng_pred1.c.

Referenced by Crc16().

7.137.4.2 `struct ng_cmdlist ng_pred1_cmds[]` [static]

Definition at line 123 of file ng_pred1.c.

7.137.4.3 `struct ng_parse_type ng_pred1_config_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_pred1_config_type_fields
}
```

Definition at line 109 of file `ng_pred1.c`.

7.137.4.4 `struct ng_parse_struct_field ng_pred1_config_type_fields[] = NG_PRED1_CONFIG_INFO` [static]

Definition at line 108 of file `ng_pred1.c`.

7.137.4.5 `ng_constructor_t ng_pred1_constructor` [static]

Definition at line 84 of file `ng_pred1.c`.

7.137.4.6 `ng_disconnect_t ng_pred1_disconnect` [static]

Definition at line 89 of file `ng_pred1.c`.

7.137.4.7 `ng_newhook_t ng_pred1_newhook` [static]

Definition at line 87 of file `ng_pred1.c`.

7.137.4.8 `ng_rcvdata_t ng_pred1_rcvdata` [static]

Definition at line 88 of file `ng_pred1.c`.

7.137.4.9 `ng_rcvmsg_t ng_pred1_rcvmsg` [static]

Definition at line 85 of file `ng_pred1.c`.

7.137.4.10 `ng_shutdown_t ng_pred1_shutdown` [static]

Definition at line 86 of file `ng_pred1.c`.

7.137.4.11 `struct ng_parse_type ng_pred1_stat_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    ng_pred1_stats_type_fields
}
```

Definition at line 117 of file `ng_pred1.c`.

7.137.4.12 `struct ng_parse_struct_field ng_pred1_stats_type_fields[] = NG_PRED1_STATS_INFO`
[static]

Definition at line 116 of file ng_pred1.c.

7.137.4.13 `struct ng_type ng_pred1_typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_PRED1_NODE_TYPE,
    .constructor =  ng_pred1_constructor,
    .rcvmsg =       ng_pred1_rcvmsg,
    .shutdown =    ng_pred1_shutdown,
    .newhook =     ng_pred1_newhook,
    .rcvdata =     ng_pred1_rcvdata,
    .disconnect =  ng_pred1_disconnect,
    .cmdlist =     ng_pred1_cmds,
}
```

Definition at line 163 of file ng_pred1.c.

7.138 /usr/src/sys/netgraph/ng_pred1.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_pred1_config](#)
- struct [ng_pred1_stats](#)

Defines

- #define [NG_PRED1_NODE_TYPE](#) "pred1"
- #define [NGM_PRED1_COOKIE](#) 1166902612
- #define [NG_PRED1_HOOK_COMP](#) "comp"
- #define [NG_PRED1_HOOK_DECOMP](#) "decomp"
- #define [NG_PRED1_CONFIG_INFO](#)
- #define [NG_PRED1_STATS_INFO](#)

Enumerations

- enum {
 - [NGM_PRED1_CONFIG](#) = 1, [NGM_PRED1_RESETRREQ](#), [NGM_PRED1_GET_STATS](#), [NGM_PRED1_CLR_STATS](#),
 - [NGM_PRED1_GETCLR_STATS](#) }

7.138.1 Define Documentation

7.138.1.1 #define NG_PRED1_CONFIG_INFO

Value:

```

{
    { "enable",      \
      { NULL }      \
    },
    { "ng_parse_uint8_type", \
      &ng_parse_uint8_type, \
    },
}
  
```

Definition at line 47 of file ng_pred1.h.

7.138.1.2 #define NG_PRED1_HOOK_COMP "comp"

Definition at line 38 of file ng_pred1.h.

Referenced by ng_pred1_newhook().

7.138.1.3 #define NG_PRED1_HOOK_DECOMP "decomp"

Definition at line 39 of file ng_pred1.h.

Referenced by ng_pred1_newhook().

7.138.1.4 #define NG_PRED1_NODE_TYPE "pred1"

Definition at line 34 of file ng_pred1.h.

7.138.1.5 #define NG_PRED1_STATS_INFO

Value:

```
{
    { "FramesPlain", &ng_parse_uint64_type }, \
    { "FramesComp", &ng_parse_uint64_type }, \
    { "FramesUncomp", &ng_parse_uint64_type }, \
    { "InOctets", &ng_parse_uint64_type }, \
    { "OutOctets", &ng_parse_uint64_type }, \
    { "Errors", &ng_parse_uint64_type }, \
    { NULL }
}
```

Definition at line 63 of file ng_pred1.h.

7.138.1.6 #define NGM_PRED1_COOKIE 1166902612

Definition at line 35 of file ng_pred1.h.

Referenced by ng_pred1_rcvdata(), and ng_pred1_rcvmsg().

7.138.2 Enumeration Type Documentation

7.138.2.1 anonymous enum

Enumerator:

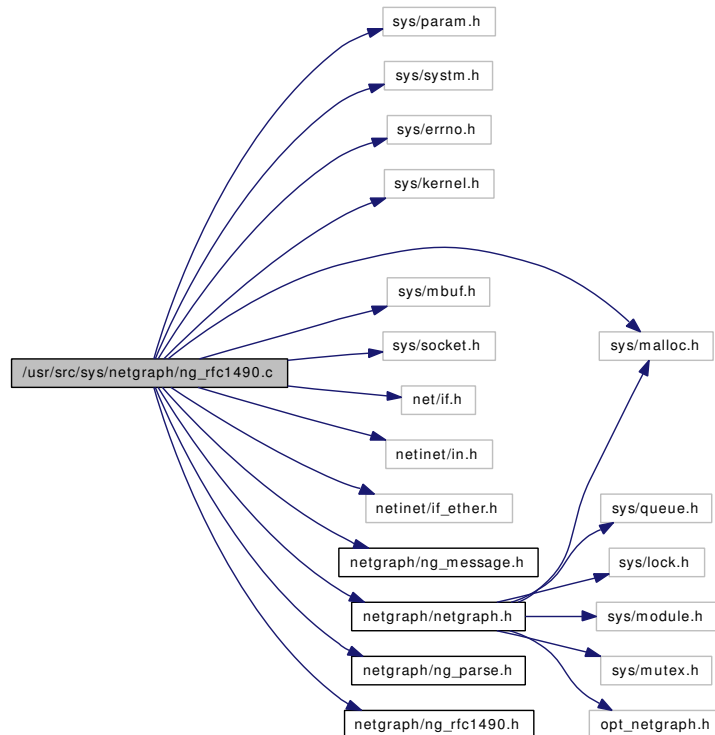
```
NGM_PRED1_CONFIG
NGM_PRED1_RESETREQ
NGM_PRED1_GET_STATS
NGM_PRED1_CLR_STATS
NGM_PRED1_GETCLR_STATS
```

Definition at line 74 of file ng_pred1.h.

7.139 /usr/src/sys/netgraph/ng_rfc1490.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet/if_ether.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_rfc1490.h>
```

Include dependency graph for ng_rfc1490.c:



Data Structures

- struct `ng_rfc1490_encap_t`

- struct [ng_rfc1490_private](#)

Defines

- #define [HDLC_UI](#) 0x03
- #define [NLPID_IP](#) 0xCC
- #define [NLPID_PPP](#) 0xCF
- #define [NLPID_SNAP](#) 0x80
- #define [NLPID_Q933](#) 0x08
- #define [NLPID_CLNP](#) 0x81
- #define [NLPID_ESIS](#) 0x82
- #define [NLPID_ISIS](#) 0x83
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)
- #define [MAX_ENCAPS_HDR](#) 8
- #define [OUICMP](#)(P, A, B, C) ((P)[0]==(A) && (P)[1]==(B) && (P)[2]==(C))

Typedefs

- typedef [ng_rfc1490_private](#) * [priv_p](#)

Enumerations

- enum { [NG_RFC1490_ENCAP_IETF_IP](#) = 1, [NG_RFC1490_ENCAP_IETF_SNAP](#), [NG_RFC1490_ENCAP_CISCO](#) }

Functions

- [NETGRAPH_INIT](#) (rfc1490,&typestruct)
- static int [ng_rfc1490_constructor](#) (node_p node)
- static int [ng_rfc1490_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ng_rfc1490_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_rfc1490_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_rfc1490_shutdown](#) (node_p node)
- static int [ng_rfc1490_disconnect](#) (hook_p hook)

Variables

- static struct [ng_rfc1490_encap_t](#) [ng_rfc1490_encaps](#) []
- static [ng_constructor_t](#) [ng_rfc1490_constructor](#)
- static [ng_rcvmsg_t](#) [ng_rfc1490_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_rfc1490_shutdown](#)
- static [ng_newhook_t](#) [ng_rfc1490_newhook](#)
- static [ng_rcvdata_t](#) [ng_rfc1490_rcvdata](#)
- static [ng_disconnect_t](#) [ng_rfc1490_disconnect](#)
- static struct [ng_cmdlist](#) [ng_rfc1490_cmds](#) []
- static struct [ng_type](#) typestruct

7.139.1 Define Documentation

7.139.1.1 **#define ERROUT(x) do { error = (x); goto done; } while (0)**

Definition at line 83 of file ng_rfc1490.c.

7.139.1.2 **#define HDLC_UI 0x03**

Definition at line 73 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata(), and ng_UI_rcvdata().

7.139.1.3 **#define MAX_ENCAPS_HDR 8**

Definition at line 304 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata(), and ng_UI_rcvdata().

7.139.1.4 **#define NLPID_CLNP 0x81**

Definition at line 79 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata().

7.139.1.5 **#define NLPID_ESIS 0x82**

Definition at line 80 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata().

7.139.1.6 **#define NLPID_IP 0xCC**

Definition at line 75 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata().

7.139.1.7 **#define NLPID_ISIS 0x83**

Definition at line 81 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata().

7.139.1.8 **#define NLPID_PPP 0xCF**

Definition at line 76 of file ng_rfc1490.c.

Referenced by ng_rfc1490_rcvdata().

7.139.1.9 **#define NLPID_Q933 0x08**

Definition at line 78 of file ng_rfc1490.c.

Referenced by `ng_rfc1490_rcvdata()`.

7.139.1.10 `#define NLPID_SNAP 0x80`

Definition at line 77 of file `ng_rfc1490.c`.

Referenced by `ng_rfc1490_rcvdata()`.

7.139.1.11 `#define OUICMP(P, A, B, C) ((P)[0]==(A) && (P)[1]==(B) && (P)[2]==(C))`

Definition at line 305 of file `ng_rfc1490.c`.

Referenced by `ng_rfc1490_rcvdata()`.

7.139.2 Typedef Documentation

7.139.2.1 `typedef struct ng_rfc1490_private* priv_p`

Definition at line 112 of file `ng_rfc1490.c`.

7.139.3 Enumeration Type Documentation

7.139.3.1 `anonymous enum`

Enumerator:

```
NG_RFC1490_ENCAP_IETF_IP  
NG_RFC1490_ENCAP_IETF_SNAP  
NG_RFC1490_ENCAP_CISCO
```

Definition at line 86 of file `ng_rfc1490.c`.

7.139.4 Function Documentation

7.139.4.1 `NETGRAPH_INIT (rfc1490, & typestruct)`

7.139.4.2 `static int ng_rfc1490_constructor (node_p node) [static]`

Definition at line 163 of file `ng_rfc1490.c`.

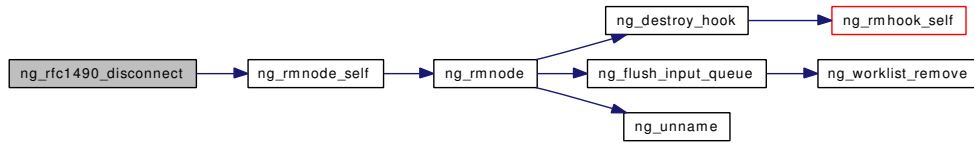
References `NG_NODE_SET_PRIVATE`, and `ng_rfc1490_encaps`.

7.139.4.3 `static int ng_rfc1490_disconnect (hook_p hook) [static]`

Definition at line 472 of file `ng_rfc1490.c`.

References `NG_HOOK_NODE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, and `ng_rmnode_self()`.

Here is the call graph for this function:



7.139.4.4 `static int ng_rfc1490_newhook (node_p node, hook_p hook, const char * name)`
`[static]`

Definition at line 185 of file `ng_rfc1490.c`.

References `NG_NODE_PRIVATE`, `NG_RFC1490_HOOK_DOWNSTREAM`, `NG_RFC1490_HOOK_ETHERNET`, `NG_RFC1490_HOOK_INET`, and `NG_RFC1490_HOOK_PPP`.

7.139.4.5 `static int ng_rfc1490_rcvdata (hook_p hook, item_p item)` `[static]`

Definition at line 308 of file `ng_rfc1490.c`.

References `ERROUT`, `HDLC_UI`, `MAX_ENCAPS_HDR`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NG_RFC1490_ENCAP_CISCO`, `NG_RFC1490_ENCAP_IETF_IP`, `NG_RFC1490_ENCAP_IETF_SNAP`, `NGI_GET_M`, `NLPID_CLNP`, `NLPID_ESIS`, `NLPID_IP`, `NLPID_ISIS`, `NLPID_PPP`, `NLPID_Q933`, `NLPID_SNAP`, and `OUICMP`.

7.139.4.6 `static int ng_rfc1490_rcvmsg (node_p node, item_p item, hook_p lasthook)` `[static]`

Definition at line 214 of file `ng_rfc1490.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ERROUT`, `ng_mesg::header`, `ng_rfc1490_encap_t::method`, `ng_rfc1490_encap_t::name`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `ng_rfc1490_encaps`, `NGI_GET_MSG`, `NGM_RFC1490_COOKIE`, `NGM_RFC1490_GET_ENCAP`, `NGM_RFC1490_SET_ENCAP`, and `ng_mesg::ng_msghdr::typecookie`.

7.139.4.7 `static int ng_rfc1490_shutdown (node_p node)` `[static]`

Definition at line 456 of file `ng_rfc1490.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.139.5 Variable Documentation

7.139.5.1 `struct ng_cmdlist ng_rfc1490_cmds[]` `[static]`

Initial value:

```

{
    {
        NGM_RFC1490_COOKIE,
        NGM_RFC1490_SET_ENCAP,
        "setencap",
        &ng_parse_string_type,
        NULL
    }
}

```

```
    },
    {
        NGM_RFC1490_COOKIE,
        NGM_RFC1490_GET_ENCAP,
        "getencap",
        NULL,
        &ng_parse_string_type
    },
    { 0 }
}
```

Definition at line 123 of file `ng_rfc1490.c`.

7.139.5.2 `ng_constructor_t ng_rfc1490_constructor` [static]

Definition at line 115 of file `ng_rfc1490.c`.

7.139.5.3 `ng_disconnect_t ng_rfc1490_disconnect` [static]

Definition at line 120 of file `ng_rfc1490.c`.

7.139.5.4 `struct ng_rfc1490_encap_t ng_rfc1490_encaps[]` [static]

Initial value:

```
{
    { NG_RFC1490_ENCAP_IETF_IP,      "ietf-ip" },
    { NG_RFC1490_ENCAP_IETF_SNAP,   "ietf-snap" },
    { NG_RFC1490_ENCAP_CISCO,      "cisco" },
    { 0, NULL },
}
```

Definition at line 97 of file `ng_rfc1490.c`.

Referenced by `ng_rfc1490_constructor()`, and `ng_rfc1490_rcvmsg()`.

7.139.5.5 `ng_newhook_t ng_rfc1490_newhook` [static]

Definition at line 118 of file `ng_rfc1490.c`.

7.139.5.6 `ng_rcvdata_t ng_rfc1490_rcvdata` [static]

Definition at line 119 of file `ng_rfc1490.c`.

7.139.5.7 `ng_rcvmsg_t ng_rfc1490_rcvmsg` [static]

Definition at line 116 of file `ng_rfc1490.c`.

7.139.5.8 `ng_shutdown_t ng_rfc1490_shutdown` [static]

Definition at line 117 of file `ng_rfc1490.c`.

7.139.5.9 struct `ng_type_t` typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_RFC1490_NODE_TYPE,
    .constructor =  ng_rfc1490_constructor,
    .rcvmsg =       ng_rfc1490_rcvmsg,
    .shutdown =     ng_rfc1490_shutdown,
    .newhook =      ng_rfc1490_newhook,
    .rcvdata =      ng_rfc1490_rcvdata,
    .disconnect =   ng_rfc1490_disconnect,
    .cmdlist =      ng_rfc1490_cmds,
}
```

Definition at line 142 of file `ng_rfc1490.c`.

7.140 /usr/src/sys/netgraph/ng_rfc1490.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define NG_RFC1490_NODE_TYPE "rfc1490"`
- `#define NGM_RFC1490_COOKIE 1086947474`
- `#define NG_RFC1490_HOOK_DOWNSTREAM "downstream"`
- `#define NG_RFC1490_HOOK_INET "inet"`
- `#define NG_RFC1490_HOOK_PPP "ppp"`
- `#define NG_RFC1490_HOOK_ETHERNET "ethernet"`

Enumerations

- `enum { NGM_RFC1490_SET_ENCAP, NGM_RFC1490_GET_ENCAP }`

7.140.1 Define Documentation

7.140.1.1 `#define NG_RFC1490_HOOK_DOWNSTREAM "downstream"`

Definition at line 52 of file `ng_rfc1490.h`.

Referenced by `ng_rfc1490_newhook()`.

7.140.1.2 `#define NG_RFC1490_HOOK_ETHERNET "ethernet"`

Definition at line 55 of file `ng_rfc1490.h`.

Referenced by `ng_rfc1490_newhook()`.

7.140.1.3 `#define NG_RFC1490_HOOK_INET "inet"`

Definition at line 53 of file `ng_rfc1490.h`.

Referenced by `ng_rfc1490_newhook()`.

7.140.1.4 `#define NG_RFC1490_HOOK_PPP "ppp"`

Definition at line 54 of file `ng_rfc1490.h`.

Referenced by `ng_rfc1490_newhook()`.

7.140.1.5 `#define NG_RFC1490_NODE_TYPE "rfc1490"`

Definition at line 48 of file `ng_rfc1490.h`.

7.140.1.6 #define NGM_RFC1490_COOKIE 1086947474

Definition at line 49 of file ng_rfc1490.h.

Referenced by ng_rfc1490_rcvmsg().

7.140.2 Enumeration Type Documentation**7.140.2.1 anonymous enum**

Enumerator:

NGM_RFC1490_SET_ENCAP

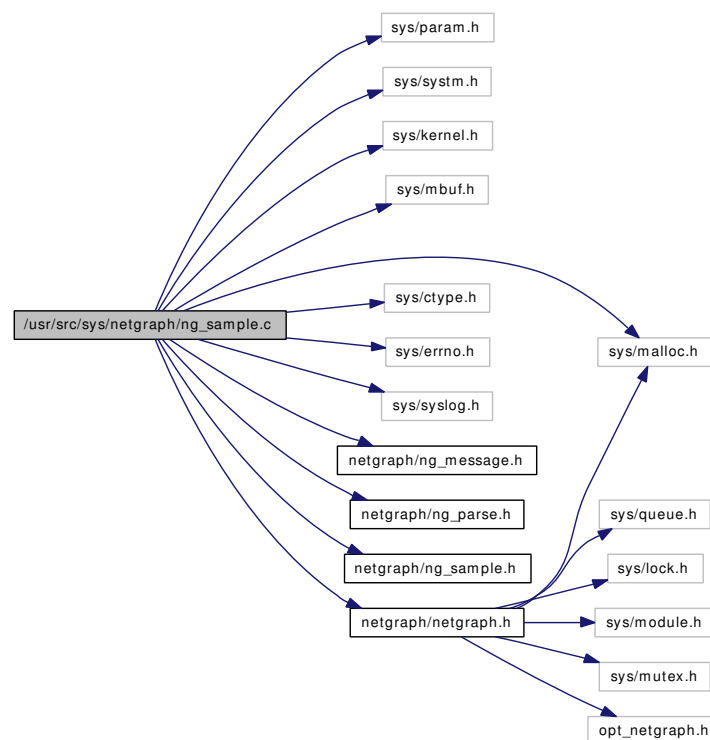
NGM_RFC1490_GET_ENCAP

Definition at line 58 of file ng_rfc1490.h.

7.141 /usr/src/sys/netgraph/ng_sample.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <sys/ctype.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_sample.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_sample.c:



Data Structures

- struct [XXX_hookinfo](#)
- struct [XXX](#)

Defines

- #define [M_NETGRAPH_XXX](#) M_NETGRAPH

Typedefs

- typedef [XXX](#) * [xxx_p](#)

Functions

- [NETGRAPH_INIT](#) ([xxx](#), &[typestruct](#))
- static int [ng_XXX_constructor](#) ([node_p](#) node)
- static int [ng_XXX_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *[name](#))
- static int [ng_XXX_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_XXX_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_XXX_shutdown](#) ([node_p](#) node)
- static int [ng_XXX_connect](#) ([hook_p](#) hook)
- static int [ng_XXX_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_XXX_constructor](#)
- static [ng_rcvmsg_t](#) [ng_XXX_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_XXX_shutdown](#)
- static [ng_newhook_t](#) [ng_XXX_newhook](#)
- static [ng_connect_t](#) [ng_XXX_connect](#)
- static [ng_rcvdata_t](#) [ng_XXX_rcvdata](#)
- static [ng_disconnect_t](#) [ng_XXX_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_XXX_stat_type_fields](#) [] = [NG_XXX_STATS_TYPE_INFO](#)
- static struct [ng_parse_type](#) [ng_XXX_stat_type](#)
- static struct [ng_cmdlist](#) [ng_XXX_cmdlist](#) []
- static struct [ng_type](#) [typestruct](#)

7.141.1 Define Documentation

7.141.1.1 #define [M_NETGRAPH_XXX](#) M_NETGRAPH

Definition at line 63 of file [ng_sample.c](#).

7.141.2 Typedef Documentation

7.141.2.1 typedef struct [XXX](#)* [xxx_p](#)

Definition at line 139 of file [ng_sample.c](#).

7.141.3 Function Documentation

7.141.3.1 NETGRAPH_INIT (*xxx*, & *typestruct*)

7.141.3.2 static int ng_XXX_connect (*hook_p hook*) [static]

Definition at line 450 of file ng_sample.c.

References XXX_hookinfo::dlci, XXX_hookinfo::hook, NG_HOOK_FORCE_QUEUE, NG_HOOK_PEER, and NG_HOOK_PRIVATE.

7.141.3.3 static int ng_XXX_constructor (*node_p node*) [static]

Definition at line 151 of file ng_sample.c.

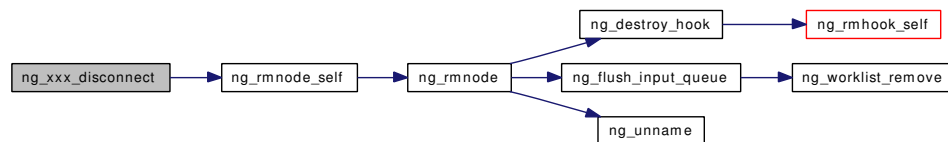
References NG_NODE_SET_PRIVATE, XXX::node, and XXX_NUM_DLCIS.

7.141.3.4 static int ng_XXX_disconnect (*hook_p hook*) [static]

Definition at line 490 of file ng_sample.c.

References XXX_hookinfo::hook, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, and ng_rmnode_self().

Here is the call graph for this function:



7.141.3.5 static int ng_XXX_newhook (*node_p node*, *hook_p hook*, *const char * name*) [static]

Definition at line 185 of file ng_sample.c.

References XXX::channel, XXX::debughook, XXX_hookinfo::dlci, XXX::downstream_hook, XXX::flags, XXX_hookinfo::hook, NG_HOOK_SET_PRIVATE, NG_NODE_PRIVATE, NG_XXX_HOOK_DEBUG, NG_XXX_HOOK_DLCI_LEADIN, NG_XXX_HOOK_DOWNSTREAM, XXX::node, and XXX_NUM_DLCIS.

7.141.3.6 static int ng_XXX_rcvdata (*hook_p hook*, *item_p item*) [static]

Definition at line 328 of file ng_sample.c.

References XXX::channel, XXX_hookinfo::channel, XXX_hookinfo::dlci, XXX::downstream_hook, XXX_hookinfo::hook, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_PRIVATE, NGI_GET_M, XXX::packets_in, XXX::packets_out, and XXX_NUM_DLCIS.

7.141.3.7 `static int ng_xxx_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 262 of file ng_sample.c.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `XXX::flags`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_XXX_COOKIE`, `NGM_XXX_GET_STATUS`, `NGM_XXX_SET_FLAG`, `XXX::node`, `XXX::packets_in`, `XXX::packets_out`, and `ng_mesg::ng_msghdr::typecookie`.

7.141.3.8 `static int ng_xxx_shutdown (node_p node)` [static]

Definition at line 420 of file ng_sample.c.

References `ng_node::nd_flags`, `NG_NODE_PRIVATE`, `NG_NODE_REVIVE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `NGF_REALLY_DIE`, and `XXX::node`.

7.141.4 Variable Documentation**7.141.4.1** `struct ng_cmdlist ng_xxx_cmdlist[]` [static]

Initial value:

```
{
    {
        NGM_XXX_COOKIE,
        NGM_XXX_GET_STATUS,
        "getstatus",
        NULL,
        &ng_xxx_stat_type,
    },
    {
        NGM_XXX_COOKIE,
        NGM_XXX_SET_FLAG,
        "setflag",
        &ng_parse_int32_type,
        NULL
    },
    { 0 }
}
```

Definition at line 88 of file ng_sample.c.

7.141.4.2 `ng_connect_t ng_xxx_connect` [static]

Definition at line 75 of file ng_sample.c.

7.141.4.3 `ng_constructor_t ng_xxx_constructor` [static]

Definition at line 71 of file ng_sample.c.

7.141.4.4 `ng_disconnect_t ng_xxx_disconnect` [static]

Definition at line 77 of file ng_sample.c.

7.141.4.5 `ng_newhook_t ng_xxx_newhook` [static]

Definition at line 74 of file ng_sample.c.

7.141.4.6 `ng_rcvdata_t ng_xxx_rcvdata` [static]

Definition at line 76 of file ng_sample.c.

7.141.4.7 `ng_rcvmsg_t ng_xxx_rcvmsg` [static]

Definition at line 72 of file ng_sample.c.

7.141.4.8 `ng_shutdown_t ng_xxx_shutdown` [static]

Definition at line 73 of file ng_sample.c.

7.141.4.9 `struct ng_parse_type ng_xxx_stat_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_xxx_stat_type_fields
}
```

Definition at line 82 of file ng_sample.c.

7.141.4.10 `struct ng_parse_struct_field ng_xxx_stat_type_fields[] = NG_XXX_STATS_TYPE_INFO` [static]

Definition at line 81 of file ng_sample.c.

7.141.4.11 `struct ng_type typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_XXX_NODE_TYPE,
    .constructor =  ng_xxx_constructor,
    .rcvmsg =       ng_xxx_rcvmsg,
    .shutdown =     ng_xxx_shutdown,
    .newhook =      ng_xxx_newhook,

    .connect =      ng_xxx_connect,
    .rcvdata =      ng_xxx_rcvdata,
    .disconnect =   ng_xxx_disconnect,
    .cmdlist =      ng_xxx_cmdlist,
}
```

Definition at line 107 of file ng_sample.c.

7.142 /usr/src/sys/netgraph/ng_sample.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngxxxstat](#)

Defines

- #define [NG_XXX_NODE_TYPE](#) "sample"
- #define [NGM_XXX_COOKIE](#) 915491374
- #define [XXX_NUM_DLCIS](#) 16
- #define [NG_XXX_HOOK_DLCI_LEADIN](#) "dlsi"
- #define [NG_XXX_HOOK_DOWNSTREAM](#) "downstream"
- #define [NG_XXX_HOOK_DEBUG](#) "debug"
- #define [NG_XXX_STATS_TYPE_INFO](#)

Enumerations

- enum { [NGM_XXX_SET_FLAG](#) = 1, [NGM_XXX_GET_STATUS](#) }

7.142.1 Define Documentation

7.142.1.1 #define [NG_XXX_HOOK_DEBUG](#) "debug"

Definition at line 62 of file `ng_sample.h`.

Referenced by `ng_XXX_newhook()`.

7.142.1.2 #define [NG_XXX_HOOK_DLCI_LEADIN](#) "dlsi"

Definition at line 60 of file `ng_sample.h`.

Referenced by `ng_XXX_newhook()`.

7.142.1.3 #define [NG_XXX_HOOK_DOWNSTREAM](#) "downstream"

Definition at line 61 of file `ng_sample.h`.

Referenced by `ng_XXX_newhook()`.

7.142.1.4 #define [NG_XXX_NODE_TYPE](#) "sample"

Definition at line 48 of file `ng_sample.h`.

7.142.1.5 #define NG_XXX_STATS_TYPE_INFO

Value:

```
{
    { "packets_in",      \
      &ng_parse_uint32_type }, \
    { "packets_out",    \
      &ng_parse_uint32_type }, \
    { NULL }
}
```

Definition at line 83 of file ng_sample.h.

7.142.1.6 #define NGM_XXX_COOKIE 915491374

Definition at line 54 of file ng_sample.h.

Referenced by ng_XXX_rcvmsg().

7.142.1.7 #define XXX_NUM_DLCIS 16

Definition at line 57 of file ng_sample.h.

Referenced by ng_XXX_constructor(), ng_XXX_newhook(), and ng_XXX_rcvdata().

7.142.2 Enumeration Type Documentation

7.142.2.1 anonymous enum

Enumerator:

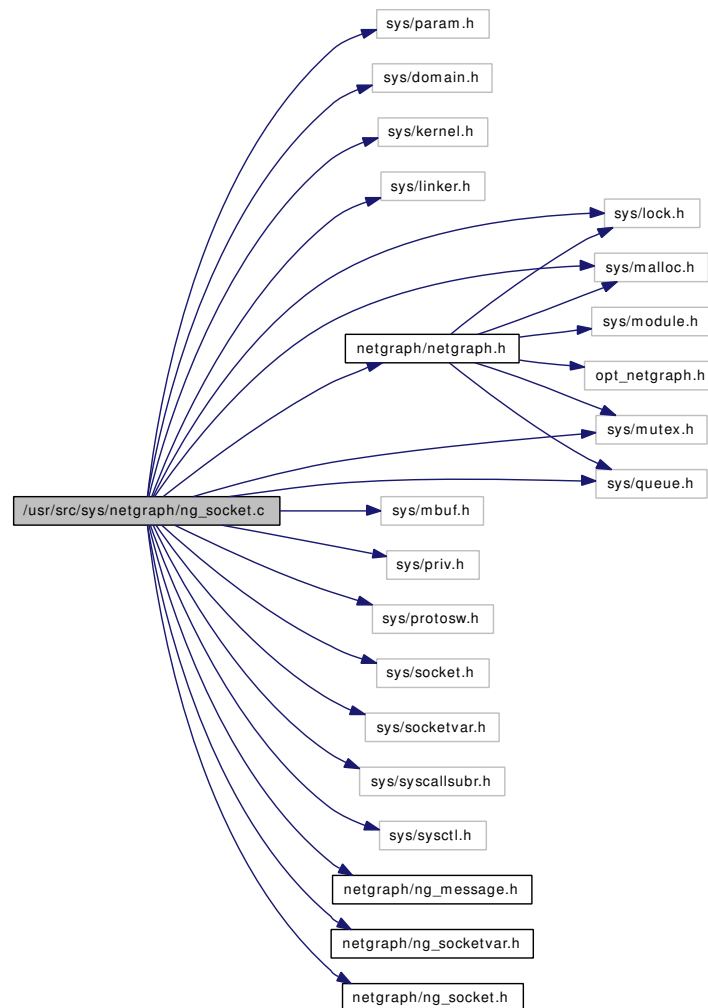
NGM_XXX_SET_FLAG
NGM_XXX_GET_STATUS

Definition at line 65 of file ng_sample.h.

7.143 /usr/src/sys/netgraph/ng_socket.c File Reference

```
#include <sys/param.h>
#include <sys/domain.h>
#include <sys/kernel.h>
#include <sys/linker.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/syscallsubr.h>
#include <sys/sysctl.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_socketvar.h>
#include <netgraph/ng_socket.h>
```

Include dependency graph for ng_socket.c:



Defines

- #define [M_NETGRAPH_PATH](#) M_NETGRAPH
- #define [M_NETGRAPH_SOCK](#) M_NETGRAPH
- #define [sotongpcb\(so\)](#) ((struct [ngpcb](#) *) (so) → so_pcb)

Functions

- static int [ng_attach_data](#) (struct socket *so)
- static int [ng_attach_cntl](#) (struct socket *so)
- static int [ng_attach_common](#) (struct socket *so, int type)
- static void [ng_detach_common](#) (struct [ngpcb](#) *pcbp, int type)
- static void [ng_socket_free_priv](#) (struct [ngsock](#) *priv)
- static int [ng_connect_data](#) (struct sockaddr *nam, struct [ngpcb](#) *pcbp)
- static int [ng_bind](#) (struct sockaddr *nam, struct [ngpcb](#) *pcbp)
- static int [ngs_mod_event](#) (module_t mod, int event, void *data)
- static void [ng_socket_item_applied](#) (void *context, int error)

- `NETGRAPH_INIT_ORDERED` (socket,&typestruct, SI_SUB_PROTO_DOMAIN, SI_ORDER_ANY)
- `SYSCTL_INT` (_net_graph, OID_AUTO, maxdgram, CTLFLAG_RW,&ngpdg_sendspace, 0,"Maximum outgoing Netgraph datagram size")
- `SYSCTL_INT` (_net_graph, OID_AUTO, recvspace, CTLFLAG_RW,&ngpdg_recvspace, 0,"Maximum space for incoming Netgraph datagrams")
- static int `ngc_attach` (struct socket *so, int proto, struct thread *td)
- static void `ngc_detach` (struct socket *so)
- static int `ngc_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td)
- static int `ngc_bind` (struct socket *so, struct sockaddr *nam, struct thread *td)
- static int `ngc_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- static int `ngd_attach` (struct socket *so, int proto, struct thread *td)
- static void `ngd_detach` (struct socket *so)
- static int `ngd_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td)
- static int `ngd_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- static int `ng_setsockaddr` (struct socket *so, struct sockaddr **addr)
- static int `ngs_constructor` (node_p nodep)
- static int `ngs_newhook` (node_p node, hook_p hook, const char *name)
- static int `ngs_connect` (hook_p hook)
- static int `ngs_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ngs_rcvdata` (hook_p hook, item_p item)
- static int `ngs_disconnect` (hook_p hook)
- static int `ngs_shutdown` (node_p node)
- static int `dummy_disconnect` (struct socket *so)
- `SYSCTL_INT` (_net_graph, OID_AUTO, family, CTLFLAG_RD, 0, AF_NETGRAPH,"")
- `SYSCTL_NODE` (_net_graph, OID_AUTO, data, CTLFLAG_RW, 0,"DATA")
- `SYSCTL_INT` (_net_graph_data, OID_AUTO, proto, CTLFLAG_RD, 0, NG_DATA,"")
- `SYSCTL_NODE` (_net_graph, OID_AUTO, control, CTLFLAG_RW, 0,"CONTROL")
- `SYSCTL_INT` (_net_graph_control, OID_AUTO, proto, CTLFLAG_RD, 0, NG_CONTROL,"")

Variables

- static `ng_constructor_t` `ngs_constructor`
- static `ng_rcvmsg_t` `ngs_rcvmsg`
- static `ng_shutdown_t` `ngs_shutdown`
- static `ng_newhook_t` `ngs_newhook`
- static `ng_connect_t` `ngs_connect`
- static `ng_rcvdata_t` `ngs_rcvdata`
- static `ng_disconnect_t` `ngs_disconnect`
- static struct `ng_type` `typestruct`
- static u_long `ngpdg_sendspace` = 20 * 1024
- static u_long `ngpdg_recvspace` = 20 * 1024
- static struct pr_usrreqs `ngc_usrreqs`
- static struct pr_usrreqs `ngd_usrreqs`
- domain `ngdomain`
- static struct protosw `ngsw` []
- domain `ngdomain`

7.143.1 Define Documentation

7.143.1.1 #define M_NETGRAPH_PATH M_NETGRAPH

Definition at line 79 of file ng_socket.c.

Referenced by ngc_send().

7.143.1.2 #define M_NETGRAPH_SOCK M_NETGRAPH

Definition at line 80 of file ng_socket.c.

Referenced by ng_attach_cntl(), and ng_socket_free_priv().

7.143.1.3 #define sotongpcb(so) ((struct ngpcb *) (so) → so_pcb)

Definition at line 156 of file ng_socket.c.

Referenced by ng_attach_cntl(), ng_setsockaddr(), ngc_attach(), ngc_bind(), ngc_detach(), ngc_send(), ngd_attach(), ngd_connect(), ngd_detach(), and ngd_send().

7.143.2 Function Documentation

7.143.2.1 static int dummy_disconnect (struct socket * so) [static]

Definition at line 1036 of file ng_socket.c.

7.143.2.2 NETGRAPH_INIT_ORDERED (socket, & tpestruct, SI_SUB_PROTO_DOMAIN, SI_ORDER_ANY)

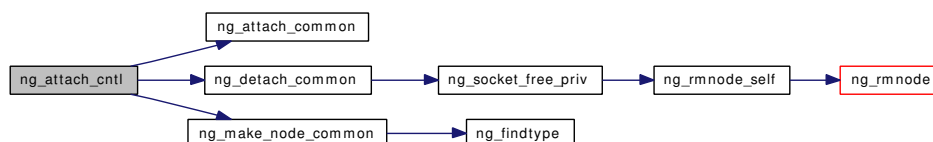
7.143.2.3 static int ng_attach_cntl (struct socket * so) [static]

Definition at line 516 of file ng_socket.c.

References M_NETGRAPH_SOCK, ng_attach_common(), NG_CONTROL, ng_detach_common(), ng_make_node_common(), NG_NODE_REF, NG_NODE_SET_PRIVATE, ngsock::refs, ngpcb::sockdata, sotongpcb, and tpestruct.

Referenced by ngc_attach().

Here is the call graph for this function:



7.143.2.4 static int ng_attach_common (struct socket * so, int type) [static]

Definition at line 566 of file ng_socket.c.

References `ngpcb::ng_socket`, `ngpdg_recvspace`, `ngpdg_sendspace`, and `ngpcb::type`.

Referenced by `ng_attach_cntl()`, and `ng_attach_data()`.

7.143.2.5 `static int ng_attach_data (struct socket *so) [static]`

Definition at line 556 of file `ng_socket.c`.

References `ng_attach_common()`, and `NG_DATA`.

Referenced by `ngd_attach()`.

Here is the call graph for this function:



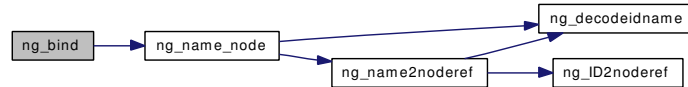
7.143.2.6 `static int ng_bind (struct sockaddr *nam, struct ngpcb *pcb) [static]`

Definition at line 772 of file `ng_socket.c`.

References `ng_name_node()`, `NG_NODESIZ`, `ngsock::node`, `sockaddr_ng::sg_data`, `sockaddr_ng::sg_len`, `ngpcb::sockdata`, and `TRAP_ERROR`.

Referenced by `ngc_bind()`.

Here is the call graph for this function:



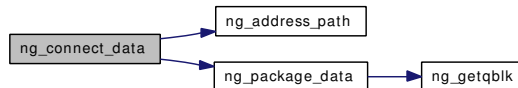
7.143.2.7 `static int ng_connect_data (struct sockaddr *nam, struct ngpcb *pcb) [static]`

Definition at line 714 of file `ng_socket.c`.

References `ngsock::datasock`, `ng_item::el_dest`, `ngsock::error`, `ngsock::mtx`, `ng_type::name`, `ng_node::nd_type`, `ng_address_path()`, `NG_FREE_ITEM`, `NG_NODE_PRIVATE`, `ng_package_data()`, `NG_SOCKET_NODE_TYPE`, `NG_WAITOK`, `ngsock::refs`, `sockaddr_ng::sg_data`, and `ngpcb::sockdata`.

Referenced by `ngd_connect()`.

Here is the call graph for this function:



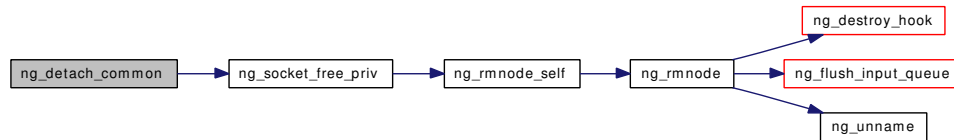
7.143.2.8 `static void ng_detach_common (struct ngpcb *pcb, int type) [static]`

Definition at line 594 of file `ng_socket.c`.

References `ngsock::ctlsock`, `ngsock::datasock`, `ngsock::mtx`, `NG_CONTROL`, `NG_DATA`, `ngpcb::ng_socket`, `ng_socket_free_priv()`, and `ngpcb::sockdata`.

Referenced by `ng_attach_cntl()`, `ngc_detach()`, and `ngd_detach()`.

Here is the call graph for this function:



7.143.2.9 `static int ng_setsockaddr (struct socket * so, struct sockaddr ** addr)` [static]

Definition at line 469 of file `ng_socket.c`.

References `NG_NODE_HAS_NAME`, `NG_NODE_NAME`, `sockaddr_ng::sg_data`, `sockaddr_ng::sg_family`, `sockaddr_ng::sg_len`, and `sotongpcb`.

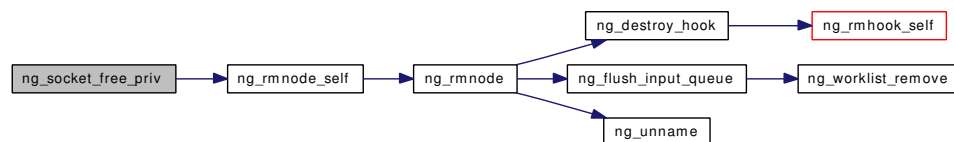
7.143.2.10 `static void ng_socket_free_priv (struct ngsock * priv)` [static]

Definition at line 624 of file `ng_socket.c`.

References `M_NETGRAPH_SOCKET`, `ngsock::mtx`, `NG_NODE_UNREF`, `ng_rmnode_self()`, `ngsock::node`, and `ngsock::refs`.

Referenced by `ng_detach_common()`, and `ngs_shutdown()`.

Here is the call graph for this function:



7.143.2.11 `static void ng_socket_item_applied (void * context, int error)` [static]

Definition at line 1024 of file `ng_socket.c`.

References `ngsock::error`, and `ngsock::mtx`.

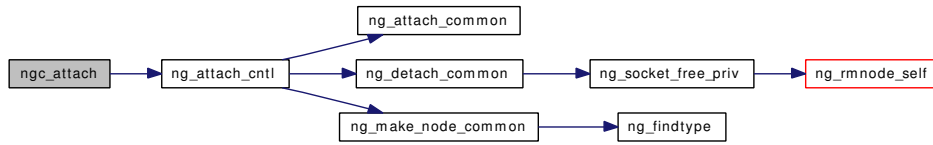
Referenced by `ngc_send()`.

7.143.2.12 `static int ngc_attach (struct socket * so, int proto, struct thread * td)` [static]

Definition at line 168 of file `ng_socket.c`.

References `ng_attach_cntl()`, and `sotongpcb`.

Here is the call graph for this function:

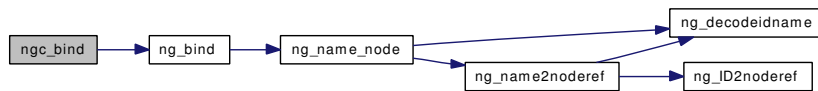


7.143.2.13 static int ngc_bind (struct socket * so, struct sockaddr * nam, struct thread * td)
[static]

Definition at line 342 of file ng_socket.c.

References ng_bind(), and sotongpcb.

Here is the call graph for this function:



7.143.2.14 static int ngc_connect (struct socket * so, struct sockaddr * nam, struct thread * td)
[static]

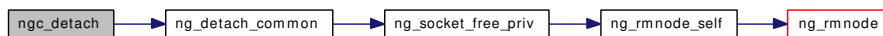
Definition at line 352 of file ng_socket.c.

7.143.2.15 static void ngc_detach (struct socket * so) [static]

Definition at line 182 of file ng_socket.c.

References NG_CONTROL, ng_detach_common(), and sotongpcb.

Here is the call graph for this function:

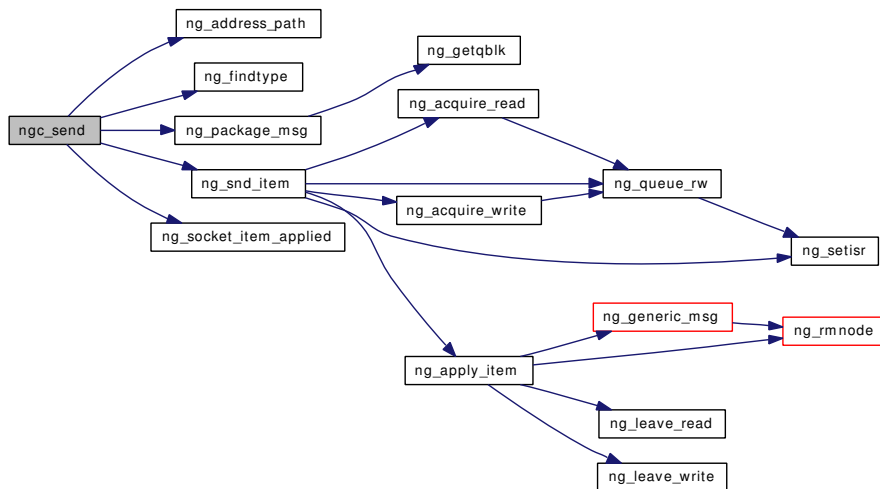


7.143.2.16 static int ngc_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * addr, struct mbuf * control, struct thread * td) [static]

Definition at line 191 of file ng_socket.c.

References ng_item::apply, ng_item::context, ng_item::el_dest, ngsock::error, M_NETGRAPH_PATH, ngsock::mtx, ng_type::name, ng_node::nd_ID, ng_node::nd_type, ng_address_path(), ng_findtype(), NG_NODE_PRIVATE, ng_package_msg(), NG_PROGRESS, ng_snd_item(), ng_socket_item_applied(), NG_TYPESIZ, NG_VERSION, NGM_GENERIC_COOKIE, NGM_MKPEER, ngsock::node, SAVE_LINE, sockaddr_ng::sg_data, sockaddr_ng::sg_len, ngpcb::sockdata, sotongpcb, and ngm_mkpeer::type.

Here is the call graph for this function:

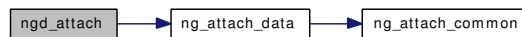


7.143.2.17 `static int ngd_attach (struct socket * so, int proto, struct thread * td)` [static]

Definition at line 367 of file `ng_socket.c`.

References `ng_attach_data()`, and `sotongpcb`.

Here is the call graph for this function:

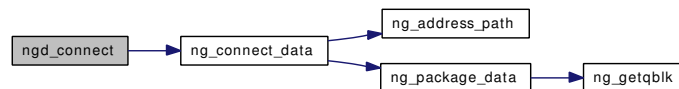


7.143.2.18 `static int ngd_connect (struct socket * so, struct sockaddr * nam, struct thread * td)` [static]

Definition at line 456 of file `ng_socket.c`.

References `ng_connect_data()`, and `sotongpcb`.

Here is the call graph for this function:

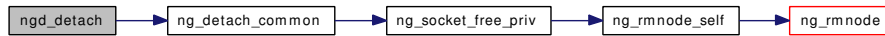


7.143.2.19 `static void ngd_detach (struct socket * so)` [static]

Definition at line 377 of file `ng_socket.c`.

References `NG_DATA`, `ng_detach_common()`, and `sotongpcb`.

Here is the call graph for this function:



7.143.2.20 `static int ngd_send (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td)` [static]

Definition at line 386 of file ng_socket.c.

References ng_findhook(), NG_HOOKSIZ, NG_NODE_NUMHOOKS, NG_SEND_DATA_FLAGS, NG_WAITOK, ngsock::node, sockaddr_ng::sg_data, sockaddr_ng::sg_len, ngpcb::sockdata, and sotongpcb.

Here is the call graph for this function:



7.143.2.21 `static int ngs_connect (hook_p hook)` [static]

Definition at line 818 of file ng_socket.c.

References ngsock::datasock, NG_HOOK_NODE, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ngpcb::ng_socket, and ngsock::node.

7.143.2.22 `static int ngs_constructor (node_p nodep)` [static]

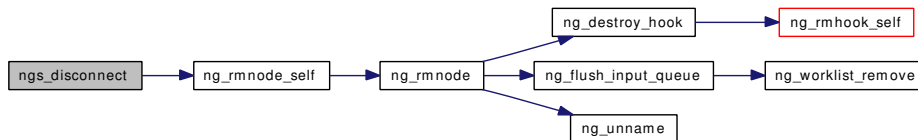
Definition at line 798 of file ng_socket.c.

7.143.2.23 `static int ngs_disconnect (hook_p hook)` [static]

Definition at line 977 of file ng_socket.c.

References ngsock::datasock, ngsock::flags, NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), ngpcb::ng_socket, NGS_FLAG_NOLINGER, and ngsock::node.

Here is the call graph for this function:



7.143.2.24 `static int ngs_mod_event (module_t mod, int event, void *data)` [static]

Definition at line 1109 of file ng_socket.c.

7.143.2.25 `static int ngs_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 808 of file ng_socket.c.

References NG_HOOK_SET_PRIVATE, and NG_NODE_PRIVATE.

7.143.2.26 `static int ngs_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 932 of file ng_socket.c.

References ngsock::datasock, NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NAME, NG_HOOK_NODE, NG_HOOKSIZ, NG_NODE_PRIVATE, ngpcb::ng_socket, NGI_GET_M, sockaddr_ng::sg_data, sockaddr_ng::sg_family, sockaddr_ng::sg_len, and TRAP_ERROR.

7.143.2.27 `static int ngs_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 837 of file ng_socket.c.

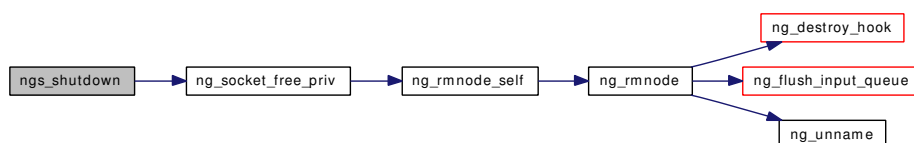
References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::ng_msghdr::cmdstr, ngsock::ctlsock, ngsock::flags, ng_mesg::ng_msghdr::flags, ng_mesg::header, NG_FREE_ITEM, NG_FREE_MSG, ng_ID_t, NG_NODE_PRIVATE, ngpcb::ng_socket, NGI_GET_MSG, NGI_RETADDR, NGM_SOCKET_CMD_LINGER, NGM_SOCKET_CMD_NOLINGER, NGM_SOCKET_COOKIE, NGS_FLAG_NOLINGER, ngsock::node, ng_mesg::ng_msghdr::token, TRAP_ERROR, and ng_mesg::ng_msghdr::typecookie.

7.143.2.28 `static int ngs_shutdown (node_p node)` [static]

Definition at line 1002 of file ng_socket.c.

References ngsock::ctlsock, ngsock::datasock, ngsock::mtx, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, ngpcb::ng_socket, ng_socket_free_priv(), and ngsock::node.

Here is the call graph for this function:



- 7.143.2.29 `SYSCTL_INT (_net_graph_control, OID_AUTO, proto, CTLFLAG_RD, 0, NG_CONTROL, "")`
- 7.143.2.30 `SYSCTL_INT (_net_graph_data, OID_AUTO, proto, CTLFLAG_RD, 0, NG_DATA, "")`
- 7.143.2.31 `SYSCTL_INT (_net_graph, OID_AUTO, family, CTLFLAG_RD, 0, AF_NETGRAPH, "")`
- 7.143.2.32 `SYSCTL_INT (_net_graph, OID_AUTO, recvspace, CTLFLAG_RW, &ngpdg_recvspace, 0, "Maximum space for incoming Netgraph datagrams")`
- 7.143.2.33 `SYSCTL_INT (_net_graph, OID_AUTO, maxdgram, CTLFLAG_RW, &ngpdg_sendspace, 0, "Maximum outgoing Netgraph datagram size")`
- 7.143.2.34 `SYSCTL_NODE (_net_graph, OID_AUTO, control, CTLFLAG_RW, 0, "CONTROL")`
- 7.143.2.35 `SYSCTL_NODE (_net_graph, OID_AUTO, data, CTLFLAG_RW, 0, "DATA")`

7.143.3 Variable Documentation

7.143.3.1 `struct pr_usrreqs ngc_usrreqs [static]`

Initial value:

```
{
    .pru_abort =          NULL,
    .pru_attach =        ngc_attach,
    .pru_bind =          ngc_bind,
    .pru_connect =       ngc_connect,
    .pru_detach =        ngc_detach,
    .pru_disconnect =    dummy_disconnect,
    .pru_peeraddr =      NULL,
    .pru_send =          ngc_send,
    .pru_shutdown =      NULL,
    .pru_sockaddr =      ng_setsockaddr,
    .pru_close =         NULL,
}
```

Definition at line 1046 of file `ng_socket.c`.

7.143.3.2 `struct pr_usrreqs ngd_usrreqs [static]`

Initial value:

```
{
    .pru_abort =          NULL,
    .pru_attach =        ngd_attach,
    .pru_bind =          NULL,
    .pru_connect =       ngd_connect,
    .pru_detach =        ngd_detach,
    .pru_disconnect =    dummy_disconnect,
    .pru_peeraddr =      NULL,
    .pru_send =          ngd_send,
    .pru_shutdown =      NULL,
    .pru_sockaddr =      ng_setsockaddr,
    .pru_close =         NULL,
}
```


Definition at line 1060 of file ng_socket.c.

7.143.3.3 struct domain **ngdomain**

Initial value:

```
{
    .dom_family =      AF_NETGRAPH,
    .dom_name =       "netgraph",
    .dom_protosw =     ngsw,
    .dom_protoswNPROTOSW = &ngsw[sizeof(ngsw) / sizeof(ngsw[0])]
}
```

Definition at line 1097 of file ng_socket.c.

7.143.3.4 struct domain **ngdomain**

Definition at line 1097 of file ng_socket.c.

7.143.3.5 u_long **ngpdg_recvspace = 20 * 1024** [static]

Definition at line 152 of file ng_socket.c.

Referenced by ng_attach_common().

7.143.3.6 u_long **ngpdg_sendspace = 20 * 1024** [static]

Definition at line 149 of file ng_socket.c.

Referenced by ng_attach_common().

7.143.3.7 **ng_connect_t ngs_connect** [static]

Definition at line 114 of file ng_socket.c.

7.143.3.8 **ng_constructor_t ngs_constructor** [static]

Definition at line 110 of file ng_socket.c.

7.143.3.9 **ng_disconnect_t ngs_disconnect** [static]

Definition at line 116 of file ng_socket.c.

7.143.3.10 **ng_newhook_t ngs_newhook** [static]

Definition at line 113 of file ng_socket.c.

7.143.3.11 **ng_rcvdata_t ngs_rcvdata** [static]

Definition at line 115 of file ng_socket.c.

7.143.3.12 `ng_rcvmsg_t ngs_rcvmsg` [static]

Definition at line 111 of file `ng_socket.c`.

7.143.3.13 `ng_shutdown_t ngs_shutdown` [static]

Definition at line 112 of file `ng_socket.c`.

7.143.3.14 `struct protosw ngsw[]` [static]**Initial value:**

```
{
{
    .pr_type =          SOCK_DGRAM,
    .pr_domain =       &ngdomain,
    .pr_protocol =     NG_CONTROL,
    .pr_flags =        PR_ATOMIC | PR_ADDR ,
    .pr_usrreqs =      &ngc_usrreqs
},
{
    .pr_type =          SOCK_DGRAM,
    .pr_domain =       &ngdomain,
    .pr_protocol =     NG_DATA,
    .pr_flags =        PR_ATOMIC | PR_ADDR,
    .pr_usrreqs =      &ngd_usrreqs
}
}
```

Definition at line 1080 of file `ng_socket.c`.

7.143.3.15 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =         NG_ABI_VERSION,
    .name =            NG_SOCKET_NODE_TYPE,
    .mod_event =       ngs_mod_event,
    .constructor =     ngs_constructor,
    .rcvmsg =          ngs_rcvmsg,
    .shutdown =        ngs_shutdown,
    .newhook =         ngs_newhook,
    .connect =         ngs_connect,
    .rcvdata =         ngs_rcvdata,
    .disconnect =      ngs_disconnect,
}
```

Definition at line 134 of file `ng_socket.c`.

7.144 /usr/src/sys/netgraph/ng_socket.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [sockaddr_ng](#)

Defines

- #define [NG_SOCKET_NODE_TYPE](#) "socket"
- #define [NGM_SOCKET_COOKIE](#) 851601233
- #define [NG_DATA](#) 1
- #define [NG_CONTROL](#) 2

Enumerations

- enum { [NGM SOCK_CMD_NOLINGER](#) = 1, [NGM SOCK_CMD_LINGER](#) }

7.144.1 Define Documentation

7.144.1.1 #define NG_CONTROL 2

Definition at line 53 of file `ng_socket.h`.

Referenced by `ng_attach_cntl()`, `ng_detach_common()`, and `ngc_detach()`.

7.144.1.2 #define NG_DATA 1

Definition at line 52 of file `ng_socket.h`.

Referenced by `ng_attach_data()`, `ng_detach_common()`, and `ngd_detach()`.

7.144.1.3 #define NG_SOCKET_NODE_TYPE "socket"

Definition at line 48 of file `ng_socket.h`.

Referenced by `ng_connect_data()`.

7.144.1.4 #define NGM_SOCKET_COOKIE 851601233

Definition at line 49 of file `ng_socket.h`.

Referenced by `ngs_rcvmsg()`.

7.144.2 Enumeration Type Documentation

7.144.2.1 anonymous enum

Enumerator:

NGM_SOCK_CMD_NOLINGER

NGM_SOCK_CMD_LINGER

Definition at line 56 of file ng_socket.h.

7.145 /usr/src/sys/netgraph/ng_socketvar.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngpcb](#)
- struct [ngsock](#)

Defines

- #define [NGS_FLAG_NOLINGER](#) 1

7.145.1 Define Documentation

7.145.1.1 #define NGS_FLAG_NOLINGER 1

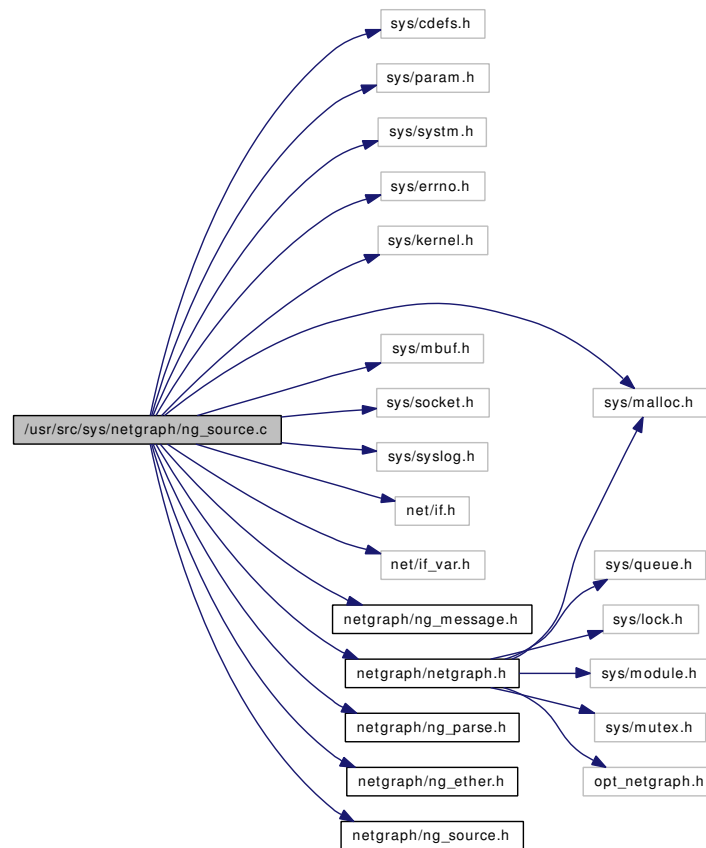
Definition at line 65 of file `ng_socketvar.h`.

Referenced by `ngs_disconnect()`, and `ngs_rcvmsg()`.

7.146 /usr/src/sys/netgraph/ng_source.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/if_var.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_ether.h>
#include <netgraph/ng_source.h>
```

Include dependency graph for ng_source.c:



Data Structures

- struct [privdata](#)

Defines

- #define [NG_SOURCE_INTR_TICKS](#) 1
- #define [NG_SOURCE_DRIVER_IFQ_MAXLEN](#) (4*1024)
- #define [NG_SOURCE_ACTIVE](#) (NGF_TYPE1)

Typedefs

- typedef [privdata](#) * [sc_p](#)

Functions

- [__FBSDID](#) ("\$FreeBSD: src/sys/netgraph/ng_source.c,v 1.27 2006/01/23 10:28:04 glebius Exp \$")
- static void [ng_source_intr](#) ([node_p](#), [hook_p](#), void *, int)
- static void [ng_source_clr_data](#) ([sc_p](#))
- static int [ng_source_start](#) ([sc_p](#), [uint64_t](#))
- static void [ng_source_stop](#) ([sc_p](#))

- static int `ng_source_send` (`sc_p`, int, int *)
- static int `ng_source_store_output_ifp` (`sc_p`, char *)
- `NETGRAPH_INIT` (source,&`ng_source_tpestruct`)
- static int `ng_source_set_autosrc` (`sc_p`, uint32_t)
- static int `ng_source_constructor` (`node_p` node)
- static int `ng_source_newhook` (`node_p` node, `hook_p` hook, const char *`name`)
- static int `ng_source_connect` (`hook_p` hook)
- static int `ng_source_rcvmsg` (`node_p` node, `item_p` item, `hook_p` lasthook)
- static int `ng_source_rcvdata` (`hook_p` hook, `item_p` item)
- static int `ng_source_rmnode` (`node_p` node)
- static int `ng_source_disconnect` (`hook_p` hook)

Variables

- static `ng_constructor_t` `ng_source_constructor`
- static `ng_rcvmsg_t` `ng_source_rcvmsg`
- static `ng_shutdown_t` `ng_source_rmnode`
- static `ng_newhook_t` `ng_source_newhook`
- static `ng_connect_t` `ng_source_connect`
- static `ng_rcvdata_t` `ng_source_rcvdata`
- static `ng_disconnect_t` `ng_source_disconnect`
- static struct `ng_parse_struct_field` `ng_source_timeval_type_fields` []
- `ng_parse_type` `ng_source_timeval_type`
- static struct `ng_parse_struct_field` `ng_source_stats_type_fields` [] = `NG_SOURCE_STATS_TYPE_INFO`
- static struct `ng_parse_type` `ng_source_stats_type`
- static struct `ng_cmdlist` `ng_source_cmds` []
- static struct `ng_type` `ng_source_tpestruct`

7.146.1 Define Documentation

7.146.1.1 #define NG_SOURCE_ACTIVE (NGF_TYPE1)

Definition at line 95 of file `ng_source.c`.

Referenced by `ng_source_intr()`, `ng_source_rcvmsg()`, `ng_source_send()`, `ng_source_start()`, and `ng_source_stop()`.

7.146.1.2 #define NG_SOURCE_DRIVER_IFQ_MAXLEN (4*1024)

Definition at line 78 of file `ng_source.c`.

Referenced by `ng_source_store_output_ifp()`.

7.146.1.3 #define NG_SOURCE_INTR_TICKS 1

Definition at line 77 of file `ng_source.c`.

Referenced by `ng_source_intr()`.

7.146.2 Typedef Documentation

7.146.2.1 typedef struct [privdata*](#) [sc_p](#)

Definition at line 92 of file `ng_source.c`.

7.146.3 Function Documentation

7.146.3.1 `__FBSDID` ("FreeBSD: src/sys/netgraph/ng_source.c, v 1.27 2006/01/23 10:28:04 glebius Exp \$")

7.146.3.2 `NETGRAPH_INIT` (`source`, & `ng_source_tpestruct`)

7.146.3.3 `static void ng_source_clr_data` (`sc_p`) [static]

Definition at line 540 of file `ng_source.c`.

References `NG_FREE_M`.

Referenced by `ng_source_rcvmsg()`, and `ng_source_rmnode()`.

7.146.3.4 `static int ng_source_connect` (`hook_p hook`) [static]

Definition at line 255 of file `ng_source.c`.

References `NG_HOOK_NODE`, `NG_MKMESSAGE`, `NG_NODE_ID`, `NG_NODE_PRIVATE`, `NG_PEER_NODE`, `NG_SEND_MSG_ID`, `NGM_ETHER_COOKIE`, `NGM_ETHER_GET_IFNAME`, and `ng_async_private::node`.

7.146.3.5 `static int ng_source_constructor` (`node_p node`) [static]

Definition at line 215 of file `ng_source.c`.

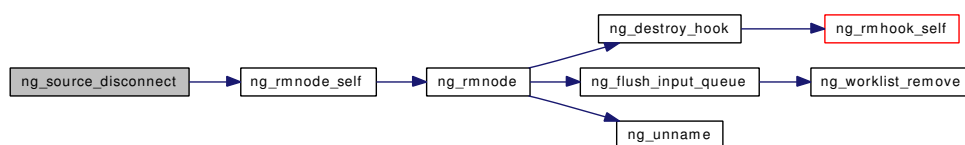
References `ng_callout_init`, and `NG_NODE_SET_PRIVATE`.

7.146.3.6 `static int ng_source_disconnect` (`hook_p hook`) [static]

Definition at line 471 of file `ng_source.c`.

References `NG_HOOK_NODE`, `NG_NODE_NUMHOOKS`, `NG_NODE_PRIVATE`, and `ng_rmnode_self()`.

Here is the call graph for this function:



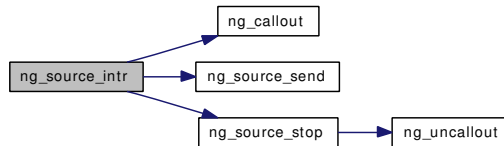
7.146.3.7 static void ng_source_intr (node_p, hook_p, void *, int) [static]

Definition at line 599 of file ng_source.c.

References ng_node::nd_flags, ng_callout(), NG_SOURCE_ACTIVE, NG_SOURCE_INTR_TICKS, ng_source_send(), ng_source_stop(), ng_async_private::node, and ng_async_private::stats.

Referenced by ng_source_start().

Here is the call graph for this function:

**7.146.3.8 static int ng_source_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 235 of file ng_source.c.

References NG_NODE_PRIVATE, NG_SOURCE_HOOK_INPUT, NG_SOURCE_HOOK_OUTPUT, and ng_async_private::stats.

7.146.3.9 static int ng_source_rcvdata (hook_p hook, item_p item) [static]

Definition at line 425 of file ng_source.c.

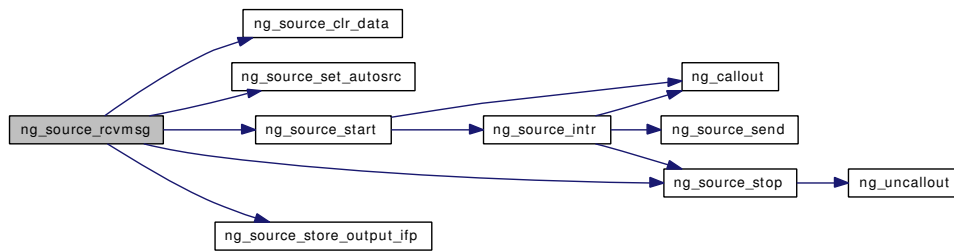
References NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NODE, NG_NODE_PRIVATE, and NGL_GET_M.

7.146.3.10 static int ng_source_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 286 of file ng_source.c.

References ng_msg::ng_msghdr::arglen, ng_msg::ng_msghdr::cmd, ng_msg::data, ng_msg::ng_msghdr::flags, ng_msg::header, ng_node::nd_flags, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NG_SOURCE_ACTIVE, ng_source_clr_data(), ng_source_set_autosrc(), ng_source_start(), ng_source_stop(), ng_source_store_output_ifp(), NGF_RESP, NGL_GET_MSG, NGM_ETHER_COOKIE, NGM_ETHER_GET_IFNAME, NGM_SOURCE_CLR_DATA, NGM_SOURCE_CLR_STATS, NGM_SOURCE_COOKIE, NGM_SOURCE_GET_STATS, NGM_SOURCE_GETCLR_STATS, NGM_SOURCE_SETIFACE, NGM_SOURCE_SETPPS, NGM_SOURCE_START, NGM_SOURCE_STOP, ng_async_private::node, ng_async_private::stats, and ng_msg::ng_msghdr::typecookie.

Here is the call graph for this function:

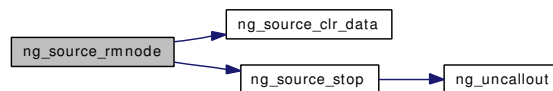


7.146.3.11 static int ng_source_rmnode (node_p node) [static]

Definition at line 454 of file ng_source.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, ng_source_clr_data(), and ng_source_stop().

Here is the call graph for this function:



7.146.3.12 static int ng_source_send (sc_p, int, int *) [static]

Definition at line 646 of file ng_source.c.

References ng_node::nd_flags, NG_SEND_DATA_ONLY, NG_SOURCE_ACTIVE, ng_async_private::node, and ng_async_private::stats.

Referenced by ng_source_intr().

7.146.3.13 static int ng_source_set_autosrc (sc_p, uint32_t) [static]

Definition at line 521 of file ng_source.c.

References ng_msg::data, NG_MKMESSAGE, NG_SEND_MSG_HOOK, NGM_ETHER_COOKIE, NGM_ETHER_SET_AUTOSRC, and ng_async_private::node.

Referenced by ng_source_rcvmsg().

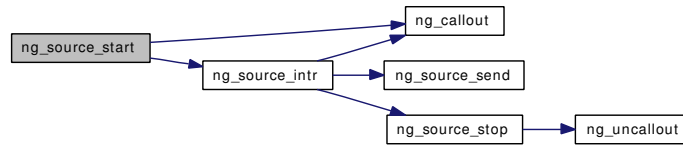
7.146.3.14 static int ng_source_start (sc_p, uint64_t) [static]

Definition at line 557 of file ng_source.c.

References ng_node::nd_flags, ng_callout(), NG_SOURCE_ACTIVE, ng_source_intr(), ng_async_private::node, and ng_async_private::stats.

Referenced by ng_source_rcvmsg().

Here is the call graph for this function:



7.146.3.15 `static void ng_source_stop(sc_p)` [static]

Definition at line 584 of file `ng_source.c`.

References `ng_node::nd_flags`, `NG_SOURCE_ACTIVE`, `ng_unccallout()`, `ng_async_private::node`, and `ng_async_private::stats`.

Referenced by `ng_source_intr()`, `ng_source_rcvmsg()`, and `ng_source_rmnode()`.

Here is the call graph for this function:



7.146.3.16 `static int ng_source_store_output_ifp(sc_p, char *)` [static]

Definition at line 487 of file `ng_source.c`.

References `NG_SOURCE_DRIVER_IFQ_MAXLEN`.

Referenced by `ng_source_rcvmsg()`.

7.146.4 Variable Documentation

7.146.4.1 `struct ng_cmdlist ng_source_cmds[]` [static]

Definition at line 134 of file `ng_source.c`.

7.146.4.2 `ng_connect_t ng_source_connect` [static]

Definition at line 102 of file `ng_source.c`.

7.146.4.3 `ng_constructor_t ng_source_constructor` [static]

Definition at line 98 of file `ng_source.c`.

7.146.4.4 `ng_disconnect_t ng_source_disconnect` [static]

Definition at line 104 of file `ng_source.c`.

7.146.4.5 `ng_newhook_t ng_source_newhook` [static]

Definition at line 101 of file `ng_source.c`.

7.146.4.6 `ng_rcvdata_t ng_source_rcvdata` [static]

Definition at line 103 of file ng_source.c.

7.146.4.7 `ng_rcvmsg_t ng_source_rcvmsg` [static]

Definition at line 99 of file ng_source.c.

7.146.4.8 `ng_shutdown_t ng_source_rmnode` [static]

Definition at line 100 of file ng_source.c.

7.146.4.9 `struct ng_parse_type ng_source_stats_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_source_stats_type_fields
}
```

Definition at line 128 of file ng_source.c.

7.146.4.10 `struct ng_parse_struct_field ng_source_stats_type_fields[] = NG_SOURCE_STATS_TYPE_INFO` [static]

Definition at line 127 of file ng_source.c.

7.146.4.11 `struct ng_parse_type ng_source_timeval_type`**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_source_timeval_type_fields
}
```

Definition at line 120 of file ng_source.c.

7.146.4.12 `struct ng_parse_struct_field ng_source_timeval_type_fields[]` [static]**Initial value:**

```
{
    { "tv_sec",          &ng_parse_int32_type  },
    { "tv_usec",        &ng_parse_int32_type  },
    { NULL }
}
```

Definition at line 115 of file ng_source.c.

7.146.4.13 struct `ng_type ng_source_type`struct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_SOURCE_NODE_TYPE,
    .constructor =  ng_source_constructor,
    .rcvmsg =      ng_source_rcvmsg,
    .shutdown =    ng_source_rmnode,
    .newhook =     ng_source_newhook,
    .connect =     ng_source_connect,
    .rcvdata =     ng_source_rcvdata,
    .disconnect =  ng_source_disconnect,
    .cmdlist =     ng_source_cmds,
}
```

Definition at line 195 of file `ng_source.c`.

7.147 /usr/src/sys/netgraph/ng_source.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_source_stats](#)

Defines

- #define [NG_SOURCE_NODE_TYPE](#) "source"
- #define [NGM_SOURCE_COOKIE](#) 1110646684
- #define [NG_SOURCE_HOOK_INPUT](#) "input"
- #define [NG_SOURCE_HOOK_OUTPUT](#) "output"
- #define [NG_SOURCE_STATS_TYPE_INFO](#)

Enumerations

- enum {
 [NGM_SOURCE_GET_STATS](#) = 1, [NGM_SOURCE_CLR_STATS](#), [NGM_SOURCE_GETCLR_STATS](#), [NGM_SOURCE_START](#),
 [NGM_SOURCE_STOP](#), [NGM_SOURCE_CLR_DATA](#), [NGM_SOURCE_SETIFACE](#), [NGM_SOURCE_SETPPS](#) }

Variables

- [ng_parse_type](#) [ng_source_timeval_type](#)

7.147.1 Define Documentation

7.147.1.1 #define [NG_SOURCE_HOOK_INPUT](#) "input"

Definition at line 50 of file [ng_source.h](#).

Referenced by [ng_source_newhook\(\)](#).

7.147.1.2 #define [NG_SOURCE_HOOK_OUTPUT](#) "output"

Definition at line 51 of file [ng_source.h](#).

Referenced by [ng_source_newhook\(\)](#).

7.147.1.3 #define [NG_SOURCE_NODE_TYPE](#) "source"

Definition at line 46 of file [ng_source.h](#).

7.147.1.4 #define NG_SOURCE_STATS_TYPE_INFO**Value:**

```

{
    { "outOctets",      &ng_parse_uint64_type }, \
    { "outFrames",     &ng_parse_uint64_type }, \
    { "queueOctets",   &ng_parse_uint32_type }, \
    { "queueFrames",  &ng_parse_uint32_type }, \
    { "maxPps",        &ng_parse_uint32_type }, \
    { "startTime",     &ng_source_timeval_type }, \
    { "endTime",       &ng_source_timeval_type }, \
    { "elapsedTime",  &ng_source_timeval_type }, \
    { "lastTime",     &ng_source_timeval_type }, \
    { NULL }
}

```

Definition at line 68 of file ng_source.h.

7.147.1.5 #define NGM_SOURCE_COOKIE 1110646684

Definition at line 47 of file ng_source.h.

Referenced by ng_source_rcvmsg().

7.147.2 Enumeration Type Documentation**7.147.2.1 anonymous enum****Enumerator:**

```

    NGM_SOURCE_GET_STATS
    NGM_SOURCE_CLR_STATS
    NGM_SOURCE_GETCLR_STATS
    NGM_SOURCE_START
    NGM_SOURCE_STOP
    NGM_SOURCE_CLR_DATA
    NGM_SOURCE_SETIFACE
    NGM_SOURCE_SETPPS

```

Definition at line 82 of file ng_source.h.

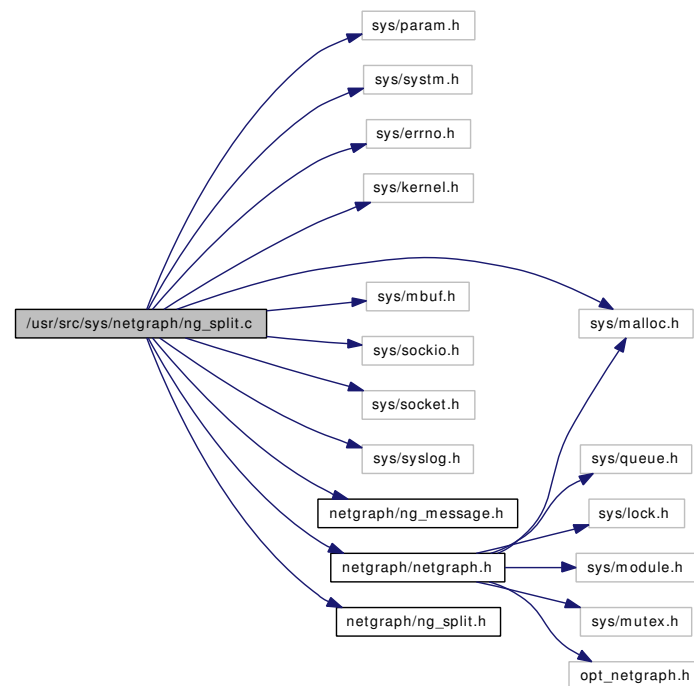
7.147.3 Variable Documentation**7.147.3.1 struct [ng_parse_type](#) [ng_source_timeval_type](#)**

Definition at line 120 of file ng_source.c.

7.148 /usr/src/sys/netgraph/ng_split.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/sockio.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_split.h>
```

Include dependency graph for ng_split.c:



Data Structures

- struct [ng_split_private](#)

Typedefs

- typedef [ng_split_private](#) * [priv_p](#)

Functions

- `NETGRAPH_INIT` (`ng_split`, &`typestruct`)
- static int `ng_split_constructor` (`node_p` `node`)
- static int `ng_split_newhook` (`node_p` `node`, `hook_p` `hook`, const char *`name`)
- static int `ng_split_rcvdata` (`hook_p` `hook`, `item_p` `item`)
- static int `ng_split_shutdown` (`node_p` `node`)
- static int `ng_split_disconnect` (`hook_p` `hook`)

Variables

- static `ng_constructor_t` `ng_split_constructor`
- static `ng_shutdown_t` `ng_split_shutdown`
- static `ng_newhook_t` `ng_split_newhook`
- static `ng_rcvdata_t` `ng_split_rcvdata`
- static `ng_disconnect_t` `ng_split_disconnect`
- static struct `ng_type` `typestruct`

7.148.1 Typedef Documentation

7.148.1.1 typedef struct `ng_split_private*` `priv_p`

Definition at line 73 of file `ng_split.c`.

7.148.2 Function Documentation

7.148.2.1 `NETGRAPH_INIT` (`ng_split`, & `typestruct`)

7.148.2.2 static int `ng_split_constructor` (`node_p` `node`) [static]

Definition at line 83 of file `ng_split.c`.

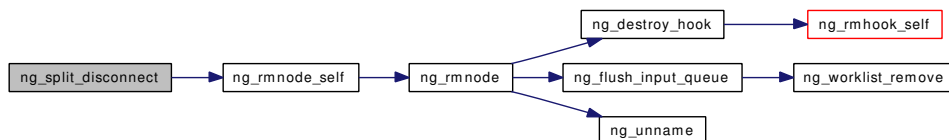
References `NG_NODE_SET_PRIVATE`, and `ng_split_private::node`.

7.148.2.3 static int `ng_split_disconnect` (`hook_p` `hook`) [static]

Definition at line 167 of file `ng_split.c`.

References `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, and `ng_rmnode_self()`.

Here is the call graph for this function:



7.148.2.4 `static int ng_split_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 104 of file ng_split.c.

References NG_HOOK_SET_PRIVATE, NG_NODE_PRIVATE, NG_SPLIT_HOOK_IN, NG_SPLIT_HOOK_MIXED, NG_SPLIT_HOOK_OUT, and ng_split_private::node.

7.148.2.5 `static int ng_split_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 130 of file ng_split.c.

References NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_HOOK_NODE, and NG_NODE_PRIVATE.

7.148.2.6 `static int ng_split_shutdown (node_p node)` [static]

Definition at line 152 of file ng_split.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_split_private::node.

7.148.3 Variable Documentation**7.148.3.1** `ng_constructor_t ng_split_constructor` [static]

Definition at line 48 of file ng_split.c.

7.148.3.2 `ng_disconnect_t ng_split_disconnect` [static]

Definition at line 52 of file ng_split.c.

7.148.3.3 `ng_newhook_t ng_split_newhook` [static]

Definition at line 50 of file ng_split.c.

7.148.3.4 `ng_rcvdata_t ng_split_rcvdata` [static]

Definition at line 51 of file ng_split.c.

7.148.3.5 `ng_shutdown_t ng_split_shutdown` [static]

Definition at line 49 of file ng_split.c.

7.148.3.6 `struct ng_type_tpestruct` [static]

Initial value:

```
{  
    .version =      NG_ABI_VERSION,  
}
```

```
.name =          NG_SPLIT_NODE_TYPE,  
.constructor =  ng_split_constructor,  
.shutdown =     ng_split_shutdown,  
.newhook =      ng_split_newhook,  
.rcvdata =      ng_split_rcvdata,  
.disconnect =   ng_split_disconnect,  
}
```

Definition at line 55 of file ng_split.c.

7.149 /usr/src/sys/netgraph/ng_split.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `NG_SPLIT_NODE_TYPE` "split"
- #define `NGM_SPLIT_COOKIE` 949409402
- #define `NG_SPLIT_HOOK_MIXED` "mixed"
- #define `NG_SPLIT_HOOK_OUT` "out"
- #define `NG_SPLIT_HOOK_IN` "in"

7.149.1 Define Documentation

7.149.1.1 #define `NG_SPLIT_HOOK_IN` "in"

Definition at line 43 of file `ng_split.h`.

Referenced by `ng_split_newhook()`.

7.149.1.2 #define `NG_SPLIT_HOOK_MIXED` "mixed"

Definition at line 41 of file `ng_split.h`.

Referenced by `ng_split_newhook()`.

7.149.1.3 #define `NG_SPLIT_HOOK_OUT` "out"

Definition at line 42 of file `ng_split.h`.

Referenced by `ng_split_newhook()`.

7.149.1.4 #define `NG_SPLIT_NODE_TYPE` "split"

Definition at line 37 of file `ng_split.h`.

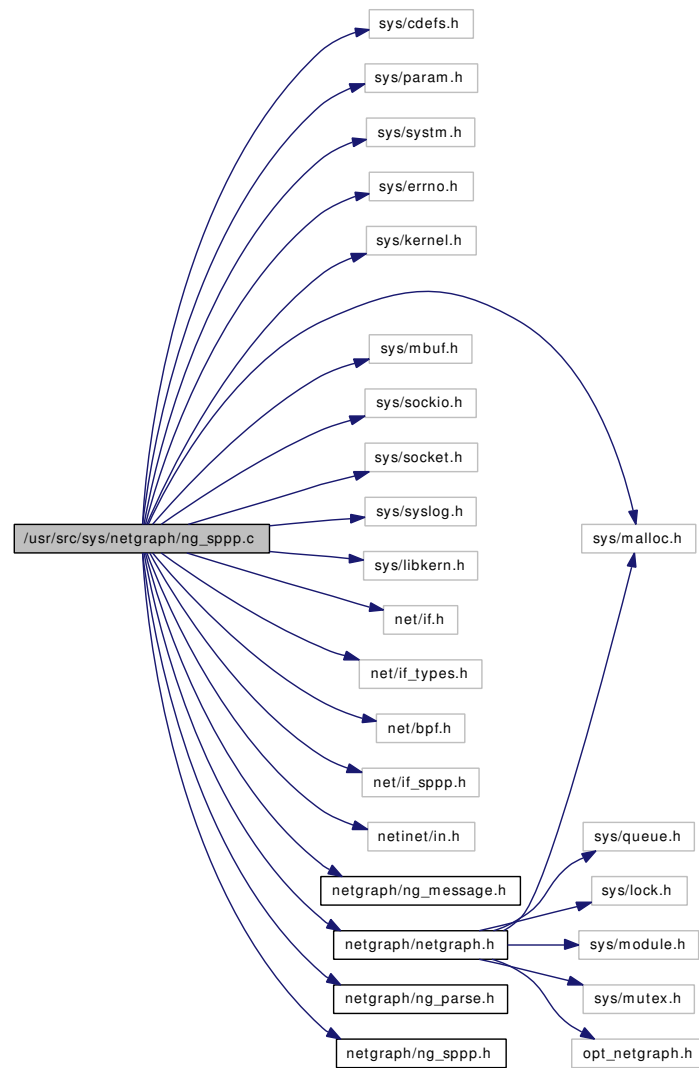
7.149.1.5 #define `NGM_SPLIT_COOKIE` 949409402

Definition at line 38 of file `ng_split.h`.

7.150 /usr/src/sys/netgraph/ng_sppp.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/sockio.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/libkern.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/bpf.h>
#include <net/if_sppp.h>
#include <netinet/in.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_sppp.h>
```

Include dependency graph for ng_sppp.c:



Data Structures

- struct [ng_sppp_private](#)

Defines

- #define [M_NETGRAPH_SPPP](#) M_NETGRAPH

Typedefs

- typedef [ng_sppp_private](#) * [priv_p](#)

Functions

- [__FBSDID](#) ("FreeBSD: src/sys/netgraph/ng_sppp.c,v 1.11 2006/12/29 13:59:50 jhb Exp \$")

- static void [ng_sppp_start](#) (struct ifnet *ifp)
- static int [ng_sppp_ioctl](#) (struct ifnet *ifp, u_long cmd, caddr_t data)
- [NETGRAPH_INIT](#) (sppp,&typestruct)
- [MODULE_DEPEND](#) (ng_sppp, sppp, 1, 1, 1)
- static __inline int [ng_sppp_get_unit](#) (int *unit)
- static __inline void [ng_sppp_free_unit](#) (int unit)
- static int [ng_sppp_constructor](#) (node_p node)
- static int [ng_sppp_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ng_sppp_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_sppp_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_sppp_shutdown](#) (node_p node)
- static int [ng_sppp_disconnect](#) (hook_p hook)

Variables

- static [ng_constructor_t](#) [ng_sppp_constructor](#)
- static [ng_rcvmsg_t](#) [ng_sppp_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_sppp_shutdown](#)
- static [ng_newhook_t](#) [ng_sppp_newhook](#)
- static [ng_rcvdata_t](#) [ng_sppp_rcvdata](#)
- static [ng_disconnect_t](#) [ng_sppp_disconnect](#)
- static struct [ng_cmdlist](#) [ng_sppp_cmds](#) []
- static struct [ng_type](#) [typestruct](#)
- static unsigned char * [ng_sppp_units](#) = NULL
- static unsigned char [ng_sppp_units_len](#) = 0
- static unsigned char [ng_units_in_use](#) = 0

7.150.1 Define Documentation

7.150.1.1 #define M_NETGRAPH_SPPP M_NETGRAPH

Definition at line 49 of file [ng_sppp.c](#).

Referenced by [ng_sppp_constructor\(\)](#), [ng_sppp_free_unit\(\)](#), [ng_sppp_get_unit\(\)](#), and [ng_sppp_shutdown\(\)](#).

7.150.2 Typedef Documentation

7.150.2.1 typedef struct [ng_sppp_private](#)* [priv_p](#)

Definition at line 59 of file [ng_sppp.c](#).

7.150.3 Function Documentation

7.150.3.1 `__FBSDID` ("FreeBSD: src/sys/netgraph/ng_sppp.c, v 1.11 2006/12/29 13:59:50 jhb Exp \$")

7.150.3.2 `MODULE_DEPEND` (ng_sppp, sppp, 1, 1, 1)

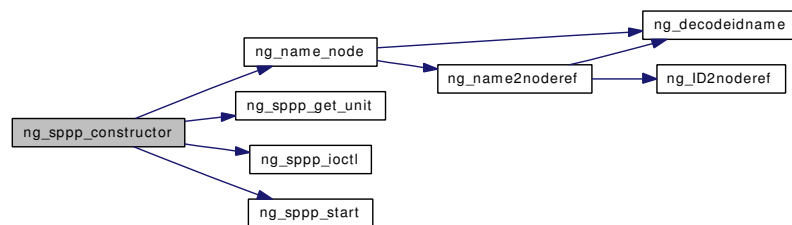
7.150.3.3 `NETGRAPH_INIT` (sppp, & typestruct)

7.150.3.4 `static int ng_sppp_constructor` (*node_p node*) [static]

Definition at line 243 of file ng_sppp.c.

References `M_NETGRAPH_SPPP`, `ng_name_node()`, `NG_NODE_SET_PRIVATE`, `ng_sppp_get_unit()`, `NG_SPPP_IFACE_NAME`, `ng_sppp_ioctl()`, and `ng_sppp_start()`.

Here is the call graph for this function:



7.150.3.5 `static int ng_sppp_disconnect` (*hook_p hook*) [static]

Definition at line 414 of file ng_sppp.c.

References `NG_HOOK_NODE`, and `NG_NODE_PRIVATE`.

7.150.3.6 `static __inline void ng_sppp_free_unit` (*int unit*) [static]

Definition at line 152 of file ng_sppp.c.

References `M_NETGRAPH_SPPP`, `ng_sppp_units`, `ng_sppp_units_len`, and `ng_units_in_use`.

Referenced by `ng_sppp_shutdown()`.

7.150.3.7 `static __inline int ng_sppp_get_unit` (*int * unit*) [static]

Definition at line 112 of file ng_sppp.c.

References `M_NETGRAPH_SPPP`, `ng_sppp_units`, `ng_sppp_units_len`, and `ng_units_in_use`.

Referenced by `ng_sppp_constructor()`.

7.150.3.8 `static int ng_sppp_ioctl` (*struct ifnet * ifp, u_long cmd, caddr_t data*) [static]

Definition at line 180 of file ng_sppp.c.

Referenced by `ng_sppp_constructor()`.

7.150.3.9 `static int ng_sppp_newhook (node_p node, hook_p hook, const char * name)`
 [static]

Definition at line 303 of file ng_sppp.c.

References NG_HOOK_SET_PRIVATE, NG_NODE_PRIVATE, and NG_SPPP_HOOK_DOWNSTREAM.

7.150.3.10 `static int ng_sppp_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 362 of file ng_sppp.c.

References NG_FREE_ITEM, NG_FREE_M, NG_HOOK_NODE, NG_NODE_PRIVATE, and NGI_GET_M.

7.150.3.11 `static int ng_sppp_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 323 of file ng_sppp.c.

References ng_mesg::ng_msghdr::cmd, ng_mesg::data, ng_mesg::header, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NGI_GET_MSG, NGM_SPPP_COOKIE, NGM_SPPP_GET_IFNAME, and ng_mesg::ng_msghdr::typecookie.

7.150.3.12 `static int ng_sppp_shutdown (node_p node)` [static]

Definition at line 395 of file ng_sppp.c.

References M_NETGRAPH_SPPP, NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, NG_NODE_UNREF, and ng_sppp_free_unit().

Here is the call graph for this function:



7.150.3.13 `static void ng_sppp_start (struct ifnet * ifp)` [static]

Definition at line 196 of file ng_sppp.c.

References NG_SEND_DATA_ONLY.

Referenced by ng_sppp_constructor().

7.150.4 Variable Documentation

7.150.4.1 `struct ng_cmdlist ng_sppp_cmds[]` [static]

Initial value:

```

{
    {
        NGM_SPPP_COOKIE,
        NGM_SPPP_GET_IFNAME,
    }
}
  
```

```
    "getifname",  
    NULL,  
    &ng_parse_string_type  
},  
{ 0 }
```

Definition at line 74 of file ng_sppp.c.

7.150.4.2 [ng_constructor_t ng_sppp_constructor](#) [static]

Definition at line 66 of file ng_sppp.c.

7.150.4.3 [ng_disconnect_t ng_sppp_disconnect](#) [static]

Definition at line 71 of file ng_sppp.c.

7.150.4.4 [ng_newhook_t ng_sppp_newhook](#) [static]

Definition at line 69 of file ng_sppp.c.

7.150.4.5 [ng_rcvdata_t ng_sppp_rcvdata](#) [static]

Definition at line 70 of file ng_sppp.c.

7.150.4.6 [ng_rcvmsg_t ng_sppp_rcvmsg](#) [static]

Definition at line 67 of file ng_sppp.c.

7.150.4.7 [ng_shutdown_t ng_sppp_shutdown](#) [static]

Definition at line 68 of file ng_sppp.c.

7.150.4.8 `unsigned char* ng_sppp_units = NULL` [static]

Definition at line 103 of file ng_sppp.c.

Referenced by [ng_sppp_free_unit\(\)](#), and [ng_sppp_get_unit\(\)](#).

7.150.4.9 `unsigned char ng_sppp_units_len = 0` [static]

Definition at line 104 of file ng_sppp.c.

Referenced by [ng_sppp_free_unit\(\)](#), and [ng_sppp_get_unit\(\)](#).

7.150.4.10 `unsigned char ng_units_in_use = 0` [static]

Definition at line 105 of file ng_sppp.c.

7.150.4.11 struct `ng_type` typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_SPPP_NODE_TYPE,
    .constructor = ng_sppp_constructor,
    .rcvmsg =       ng_sppp_rcvmsg,
    .shutdown =     ng_sppp_shutdown,
    .newhook =      ng_sppp_newhook,
    .rcvdata =      ng_sppp_rcvdata,
    .disconnect =   ng_sppp_disconnect,
    .cmdlist =      ng_sppp_cmds,
}
```

Definition at line 86 of file `ng_sppp.c`.

7.151 /usr/src/sys/netgraph/ng_sppp.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_SPPP_NODE_TYPE](#) "sppp"
- #define [NGM_SPPP_COOKIE](#) 1040804655
- #define [NG_SPPP_IFACE_NAME](#) "sppp"
- #define [NG_SPPP_HOOK_DOWNSTREAM](#) "downstream"

Enumerations

- enum { [NGM_SPPP_GET_IFNAME](#) = 1 }

7.151.1 Define Documentation

7.151.1.1 #define [NG_SPPP_HOOK_DOWNSTREAM](#) "downstream"

Definition at line 32 of file `ng_sppp.h`.

Referenced by `ng_sppp_newhook()`.

7.151.1.2 #define [NG_SPPP_IFACE_NAME](#) "sppp"

Definition at line 29 of file `ng_sppp.h`.

Referenced by `ng_sppp_constructor()`.

7.151.1.3 #define [NG_SPPP_NODE_TYPE](#) "sppp"

Definition at line 25 of file `ng_sppp.h`.

7.151.1.4 #define [NGM_SPPP_COOKIE](#) 1040804655

Definition at line 26 of file `ng_sppp.h`.

Referenced by `ng_sppp_rcvmsg()`.

7.151.2 Enumeration Type Documentation

7.151.2.1 anonymous enum

Enumerator:

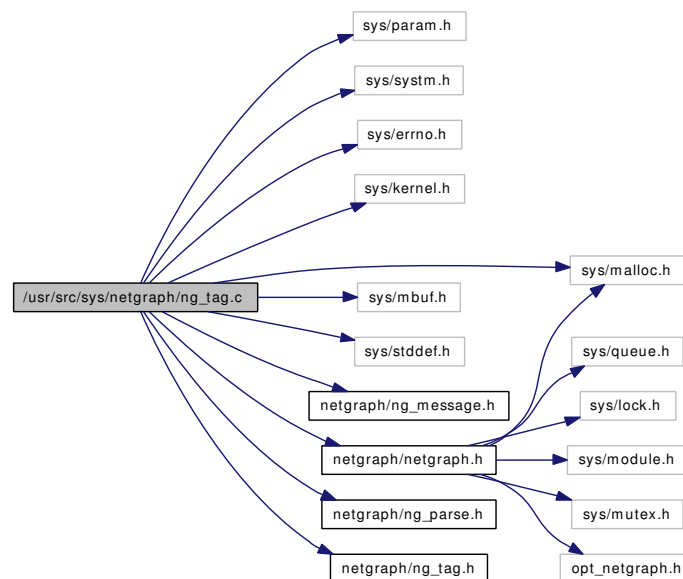
[NGM_SPPP_GET_IFNAME](#)

Definition at line 35 of file ng_sppp.h.

7.152 /usr/src/sys/netgraph/ng_tag.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/stddef.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_tag.h>
```

Include dependency graph for ng_tag.c:



Data Structures

- struct [ng_tag_hookinfo](#)

Defines

- #define [M_NETGRAPH_TAG](#) M_NETGRAPH
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef [ng_tag_hookinfo](#) * [hinfo_p](#)

Functions

- static int [ng_tag_setdata_in](#) ([hook_p](#) hook, const struct [ng_tag_hookin](#) *hp)
- static int [ng_tag_setdata_out](#) ([hook_p](#) hook, const struct [ng_tag_hookout](#) *hp)
- static int [ng_tag_hookinary_getLength](#) (const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)
- static int [ng_tag_hookoutary_getLength](#) (const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)
- [NETGRAPH_INIT](#) (tag,&typestruct)
- static int [ng_tag_constructor](#) ([node_p](#) node)
- static int [ng_tag_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_tag_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_tag_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_tag_shutdown](#) ([node_p](#) node)
- static int [ng_tag_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_tag_constructor](#)
- static [ng_rcvmsg_t](#) [ng_tag_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_tag_shutdown](#)
- static [ng_newhook_t](#) [ng_tag_newhook](#)
- static [ng_rcvdata_t](#) [ng_tag_rcvdata](#)
- static [ng_disconnect_t](#) [ng_tag_disconnect](#)
- static struct [ng_parse_type](#) [ng_tag_hookinary_type](#)
- static struct [ng_parse_type](#) [ng_tag_hookoutary_type](#)
- static struct [ng_parse_struct_field](#) [ng_tag_hookin_type_fields](#) [] = [NG_TAG_HOOKIN_TYPE_-INFO\(&ng_tag_hookinary_type\)](#)
- static struct [ng_parse_type](#) [ng_tag_hookin_type](#)
- static struct [ng_parse_struct_field](#) [ng_tag_hookout_type_fields](#) [] = [NG_TAG_HOOKOUT_-TYPE_INFO\(&ng_tag_hookoutary_type\)](#)
- static struct [ng_parse_type](#) [ng_tag_hookout_type](#)
- static struct [ng_cmdlist](#) [ng_tag_cmdlist](#) []
- static struct [ng_type](#) [typestruct](#)
- static struct [ng_tag_hookin](#) [ng_tag_default_in](#)
- static struct [ng_tag_hookout](#) [ng_tag_default_out](#)

7.152.1 Define Documentation

7.152.1.1 #define [ERROUT\(x\)](#) do { error = (x); goto done; } while (0)

Definition at line 77 of file [ng_tag.c](#).

7.152.1.2 #define M_NETGRAPH_TAG M_NETGRAPH

Definition at line 74 of file ng_tag.c.

Referenced by ng_tag_disconnect(), ng_tag_newhook(), ng_tag_setdata_in(), and ng_tag_setdata_out().

7.152.2 Typedef Documentation

7.152.2.1 typedef struct ng_tag_hookinfo* hinfo_p

Definition at line 112 of file ng_tag.c.

7.152.3 Function Documentation

7.152.3.1 NETGRAPH_INIT (tag, & typestruct)

7.152.3.2 static int ng_tag_constructor (node_p node) [static]

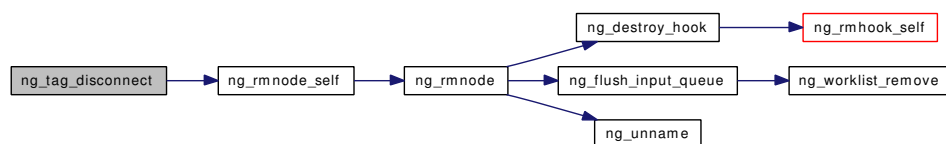
Definition at line 291 of file ng_tag.c.

7.152.3.3 static int ng_tag_disconnect (hook_p hook) [static]

Definition at line 607 of file ng_tag.c.

References M_NETGRAPH_TAG, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, and ng_rmnode_self().

Here is the call graph for this function:



7.152.3.4 static int ng_tag_hookinary_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]

Definition at line 128 of file ng_tag.c.

References ng_tag_hookin::tag_data, and ng_tag_hookin::tag_len.

7.152.3.5 static int ng_tag_hookoutary_getLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]

Definition at line 139 of file ng_tag.c.

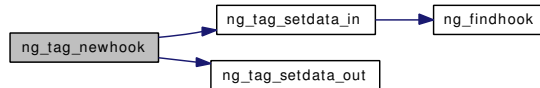
References ng_tag_hookout::tag_data, and ng_tag_hookout::tag_len.

7.152.3.6 static int ng_tag_newhook (**node_p** node, **hook_p** hook, const char * name) [static]

Definition at line 300 of file ng_tag.c.

References M_NETGRAPH_TAG, NG_HOOK_SET_PRIVATE, ng_tag_default_in, ng_tag_default_out, ng_tag_setdata_in(), and ng_tag_setdata_out().

Here is the call graph for this function:



7.152.3.7 static int ng_tag_rcvdata (**hook_p** hook, **item_p** item) [static]

Definition at line 502 of file ng_tag.c.

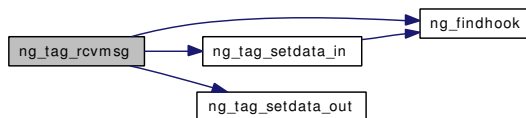
References NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_HOOK_PRIVATE, NGI_M, ng_bpf_hookstat::recvFrames, ng_bpf_hookstat::recvMatchFrames, ng_bpf_hookstat::recvMatchOctets, ng_bpf_hookstat::recvOctets, ng_bpf_hookinfo::stats, ng_bpf_hookstat::xmitFrames, and ng_bpf_hookstat::xmitOctets.

7.152.3.8 static int ng_tag_rcvmsg (**node_p** node, **item_p** item, **hook_p** lasthook) [static]

Definition at line 346 of file ng_tag.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERROUT, ng_mesg::header, ng_findhook(), NG_FREE_MSG, NG_HOOK_PRIVATE, NG_MKRESPONSE, NG_RESPOND_MSG, NG_TAG_HOOKIN_SIZE, NG_TAG_HOOKOUT_SIZE, ng_tag_setdata_in(), ng_tag_setdata_out(), NGI_GET_MSG, NGM_TAG_COOKIE, NGM_TAG_GET_HOOKIN, NGM_TAG_GET_HOOKOUT, NGM_TAG_SET_HOOKIN, NGM_TAG_SET_HOOKOUT, ng_tag_hookout::tag_len, ng_tag_hookin::tag_len, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



7.152.3.9 static int ng_tag_setdata_in (**hook_p** hook, const struct **ng_tag_hookin** * hp) [static]

Definition at line 643 of file ng_tag.c.

References M_NETGRAPH_TAG, ng_findhook(), NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_TAG_HOOKIN_SIZE, and ng_tag_hookin::tag_len.

Referenced by ng_tag_newhook(), and ng_tag_rcvmsg().

Here is the call graph for this function:



7.152.3.10 `static int ng_tag_setdata_out` ([hook_p](#) *hook*, `const struct ng_tag_hookout * hp`)
[static]

Definition at line 693 of file `ng_tag.c`.

References `M_NETGRAPH_TAG`, `NG_HOOK_PRIVATE`, `NG_TAG_HOOKOUT_SIZE`, and `ng_tag_hookout::tag_len`.

Referenced by `ng_tag_newhook()`, and `ng_tag_rcvmsg()`.

7.152.3.11 `static int ng_tag_shutdown` ([node_p](#) *node*) [static]

Definition at line 595 of file `ng_tag.c`.

References `NG_NODE_UNREF`.

7.152.4 Variable Documentation

7.152.4.1 `struct ng_cmdlist ng_tag_cmdlist[]` [static]

Definition at line 186 of file `ng_tag.c`.

7.152.4.2 `ng_constructor_t ng_tag_constructor` [static]

Definition at line 115 of file `ng_tag.c`.

7.152.4.3 `struct ng_tag_hookin ng_tag_default_in` [static]

Initial value:

```

{
    { '\0' },
    { '\0' },
    { '\0' },
    0,
    0,
    0,
    0
}
  
```

Definition at line 267 of file `ng_tag.c`.

Referenced by `ng_tag_newhook()`.

7.152.4.4 `struct ng_tag_hookout ng_tag_default_out` [static]

Initial value:

```
{
    { '\0' },
    0,
    0,
    0
}
```

Definition at line 278 of file ng_tag.c.

Referenced by ng_tag_newhook().

7.152.4.5 [ng_disconnect_t ng_tag_disconnect](#) [static]

Definition at line 120 of file ng_tag.c.

7.152.4.6 [struct ng_parse_type ng_tag_hookin_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_tag_hookin_type_fields
}
```

Definition at line 162 of file ng_tag.c.

7.152.4.7 [struct ng_parse_struct_field ng_tag_hookin_type_fields\[\] = NG_TAG_HOOKIN_TYPE_INFO\(&ng_tag_hookinary_type\)](#) [static]

Definition at line 161 of file ng_tag.c.

7.152.4.8 [struct ng_parse_type ng_tag_hookinary_type](#) [static]

Initial value:

```
{
    &ng_parse_bytearray_type,
    &ng_tag_hookinary_getLength
}
```

Definition at line 149 of file ng_tag.c.

7.152.4.9 [struct ng_parse_type ng_tag_hookout_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_tag_hookout_type_fields
}
```

Definition at line 170 of file ng_tag.c.

7.152.4.10 `struct ng_parse_struct_field ng_tag_hookout_type_fields[] =`
`NG_TAG_HOOKOUT_TYPE_INFO(&ng_tag_hookoutary_type)` [static]

Definition at line 169 of file ng_tag.c.

7.152.4.11 `struct ng_parse_type ng_tag_hookoutary_type` [static]

Initial value:

```
{
    &ng_parse_bytearray_type,
    &ng_tag_hookoutary_getLength
}
```

Definition at line 154 of file ng_tag.c.

7.152.4.12 `ng_newhook_t ng_tag_newhook` [static]

Definition at line 118 of file ng_tag.c.

7.152.4.13 `ng_rcvdata_t ng_tag_rcvdata` [static]

Definition at line 119 of file ng_tag.c.

7.152.4.14 `ng_rcvmsg_t ng_tag_rcvmsg` [static]

Definition at line 116 of file ng_tag.c.

7.152.4.15 `ng_shutdown_t ng_tag_shutdown` [static]

Definition at line 117 of file ng_tag.c.

7.152.4.16 `struct ng_type typestruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_TAG_NODE_TYPE,
    .constructor = ng_tag_constructor,
    .rcvmsg =      ng_tag_rcvmsg,
    .shutdown =    ng_tag_shutdown,
    .newhook =     ng_tag_newhook,
    .rcvdata =     ng_tag_rcvdata,
    .disconnect =  ng_tag_disconnect,
    .cmdlist =     ng_tag_cmdlist,
}
```

Definition at line 242 of file ng_tag.c.

7.153 /usr/src/sys/netgraph/ng_tag.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_tag_hookin](#)
- struct [ng_tag_hookout](#)

Defines

- #define [NG_TAG_NODE_TYPE](#) "tag"
- #define [NGM_TAG_COOKIE](#) 1149771193
- #define [NG_TAG_HOOKIN_SIZE](#)(taglen) (sizeof(struct [ng_tag_hookin](#)) + (taglen))
- #define [NG_TAG_HOOKOUT_SIZE](#)(taglen) (sizeof(struct [ng_tag_hookout](#)) + (taglen))
- #define [NG_TAG_HOOKIN_TYPE_INFO](#)(tdtype)
- #define [NG_TAG_HOOKOUT_TYPE_INFO](#)(tdtype)

Enumerations

- enum { [NGM_TAG_SET_HOOKIN](#) = 1, [NGM_TAG_GET_HOOKIN](#), [NGM_TAG_SET_HOOKOUT](#), [NGM_TAG_GET_HOOKOUT](#) }

7.153.1 Define Documentation

7.153.1.1 #define [NG_TAG_HOOKIN_SIZE](#)(taglen) (sizeof(struct [ng_tag_hookin](#)) + (taglen))

Definition at line 64 of file [ng_tag.h](#).

Referenced by [ng_tag_rcvmsg\(\)](#), and [ng_tag_setdata_in\(\)](#).

7.153.1.2 #define [NG_TAG_HOOKIN_TYPE_INFO](#)(tdtype)

Value:

```

{
    \
    { "thisHook",      &ng_parse_hookbuf_type }, \
    { "ifMatch",      &ng_parse_hookbuf_type }, \
    { "ifNotMatch",   &ng_parse_hookbuf_type }, \
    { "strip",        &ng_parse_uint8_type }, \
    { "tag_cookie",   &ng_parse_uint32_type }, \
    { "tag_id",       &ng_parse_uint16_type }, \
    { "tag_len",      &ng_parse_uint16_type }, \
    { "tag_data",     (tdtype) }, \
    { NULL }
}

```

Definition at line 71 of file [ng_tag.h](#).

7.153.1.3 #define NG_TAG_HOOKOUT_SIZE(taglen) (sizeof(struct ng_tag_hookout) + (taglen))

Definition at line 67 of file ng_tag.h.

Referenced by ng_tag_rcvmsg(), and ng_tag_setdata_out().

7.153.1.4 #define NG_TAG_HOOKOUT_TYPE_INFO(tdtype)

Value:

```
{
    \
    { "thisHook",      &ng_parse_hookbuf_type  }, \
    { "tag_cookie",   &ng_parse_uint32_type  }, \
    { "tag_id",       &ng_parse_uint16_type  }, \
    { "tag_len",      &ng_parse_uint16_type  }, \
    { "tag_data",     (tdtype)                }, \
    { NULL }
}
```

Definition at line 83 of file ng_tag.h.

7.153.1.5 #define NG_TAG_NODE_TYPE "tag"

Definition at line 34 of file ng_tag.h.

7.153.1.6 #define NGM_TAG_COOKIE 1149771193

Definition at line 35 of file ng_tag.h.

Referenced by ng_tag_rcvmsg().

7.153.2 Enumeration Type Documentation**7.153.2.1 anonymous enum**

Enumerator:

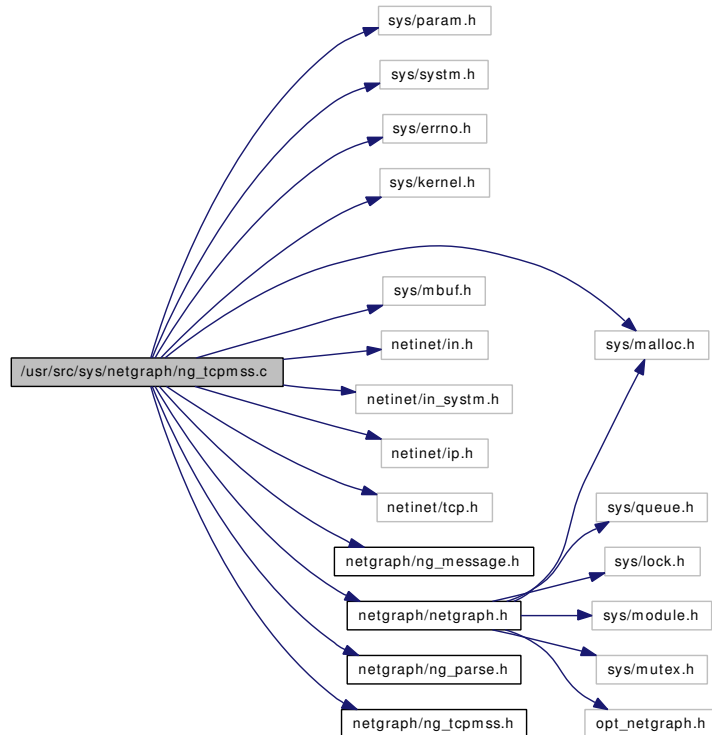
```
NGM_TAG_SET_HOOKIN
NGM_TAG_GET_HOOKIN
NGM_TAG_SET_HOOKOUT
NGM_TAG_GET_HOOKOUT
```

Definition at line 118 of file ng_tag.h.

7.154 /usr/src/sys/netgraph/ng_tcpmss.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_tcpmss.h>
```

Include dependency graph for ng_tcpmss.c:



Data Structures

- struct [hpriv_p](#)

Defines

- #define [ERROUT\(x\)](#) { error = (x); goto done; }
- #define [M_CHECK\(length\)](#)
- #define [TCPMSS_ADJUST_CHECKSUM\(acc, cksum\)](#)

Functions

- static int [correct_mss](#) (struct tcphdr *, int, uint16_t, int)
- [NETGRAPH_INIT](#) (tcpmss,&ng_tcpmss_typestruct)
- static int [ng_tcpmss_constructor](#) (node_p node)
- static int [ng_tcpmss_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ng_tcpmss_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_tcpmss_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_tcpmss_disconnect](#) (hook_p hook)

Variables

- static [ng_constructor_t](#) [ng_tcpmss_constructor](#)
- static [ng_rcvmsg_t](#) [ng_tcpmss_rcvmsg](#)
- static [ng_newhook_t](#) [ng_tcpmss_newhook](#)
- static [ng_rcvdata_t](#) [ng_tcpmss_rcvdata](#)
- static [ng_disconnect_t](#) [ng_tcpmss_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_tcpmss_hookstat_type_fields](#) [] = NG_TCPMSS_HOOKSTAT_INFO
- static struct [ng_parse_type](#) [ng_tcpmss_hookstat_type](#)
- static struct [ng_parse_struct_field](#) [ng_tcpmss_config_type_fields](#) [] = NG_TCPMSS_CONFIG_INFO
- static struct [ng_parse_type](#) [ng_tcpmss_config_type](#)
- static struct [ng_cmdlist](#) [ng_tcpmss_cmds](#) []
- static struct [ng_type](#) [ng_tcpmss_typestruct](#)

7.154.1 Define Documentation

7.154.1.1 #define ERROUT(x) { error = (x); goto done; }

Definition at line 143 of file ng_tcpmss.c.

7.154.1.2 #define M_CHECK(length)

Value:

```
do {
    pullup_len += length;
    if ((m)->m_pkthdr.len < pullup_len)
        goto send;
    if ((m)->m_len < pullup_len &&
        ((m) = m_pullup(m), pullup_len)) == NULL)
        ERROUT(ENOBUFS);
} while (0)
```

7.154.1.3 #define TCPMSS_ADJUST_CHECKSUM(acc, cksum)

Value:

```
do {
    \
    acc += cksum;
    \
    if (acc < 0) {
    \
        acc = -acc;
    \
        acc = (acc >> 16) + (acc & 0xffff);
    \
        acc += acc >> 16;
    \
        cksum = (u_short) ~acc;
    \
    } else {
    \
        acc = (acc >> 16) + (acc & 0xffff);
    \
        acc += acc >> 16;
    \
        cksum = (u_short) acc;
    \
    }
} while (0);
```

Definition at line 394 of file ng_tcpmss.c.

Referenced by correct_mss().

7.154.2 Function Documentation

7.154.2.1 static int correct_mss (struct tephdr *, int, uint16_t, int) [static]

Definition at line 409 of file ng_tcpmss.c.

References TCPMSS_ADJUST_CHECKSUM.

Referenced by ng_tcpmss_rcvdata().

7.154.2.2 NETGRAPH_INIT (tcpmss, & ng_tcpmss_tpestruct)

7.154.2.3 static int ng_tcpmss_constructor (node_p node) [static]

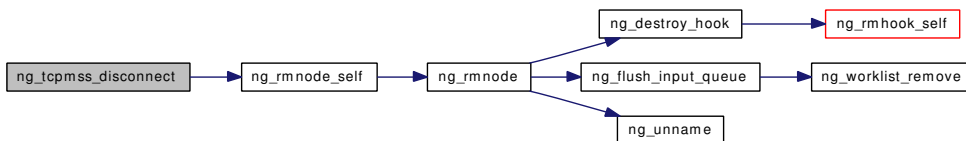
Definition at line 149 of file ng_tcpmss.c.

7.154.2.4 static int ng_tcpmss_disconnect (hook_p hook) [static]

Definition at line 362 of file ng_tcpmss.c.

References NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_NODE_NUMHOOKS, ng_rmnode_self(), and hpriv_p::outHook.

Here is the call graph for this function:



7.154.2.5 static int ng_tcpmss_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 158 of file ng_tcpmss.c.

References NG_HOOK_SET_PRIVATE.

7.154.2.6 static int ng_tcpmss_rcvdata (hook_p hook, item_p item) [static]

Definition at line 265 of file ng_tcpmss.c.

References correct_mss(), ERROUT, ng_tcpmss_hookstat::FixedPkts, M_CHECK, ng_tcpmss_hookstat::maxMSS, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_PRIVATE, NGL_GET_M, ng_tcpmss_hookstat::Octets, hpriv_p::outHook, ng_tcpmss_hookstat::Packets, hpriv_p::stats, and ng_tcpmss_hookstat::SYNPkts.

Here is the call graph for this function:

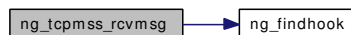


7.154.2.7 static int ng_tcpmss_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 176 of file ng_tcpmss.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERROUT, ng_mesg::header, ng_tcpmss_config::inHook, ng_tcpmss_config::maxMSS, ng_tcpmss_hookstat::maxMSS, ng_findhook(), NG_FREE_MSG, NG_HOOK_PRIVATE, NG_HOOKSIZ, NG_MKRESPONSE, NG_RESPOND_MSG, NGL_GET_MSG, NGM_TCPMSS_CLR_STATS, NGM_TCPMSS_CONFIG, NGM_TCPMSS_COOKIE, NGM_TCPMSS_GET_STATS, NGM_TCPMSS_GETCLR_STATS, hpriv_p::outHook, ng_tcpmss_config::outHook, hpriv_p::stats, and ng_mesg::ng_msghdr::typecookie.

Here is the call graph for this function:



7.154.3 Variable Documentation

7.154.3.1 struct ng_cmdlist ng_tcpmss_cmds[] [static]

Definition at line 97 of file ng_tcpmss.c.

7.154.3.2 struct ng_parse_type ng_tcpmss_config_type [static]

Initial value:

```

{
    &ng_parse_struct_type,
    ng_tcpmss_config_type_fields
}
  
```

Definition at line 91 of file ng_tcpmss.c.

7.154.3.3 `struct ng_parse_struct_field ng_tcpmss_config_type_fields[] =`
`NG_TCPMSS_CONFIG_INFO [static]`

Definition at line 90 of file ng_tcpmss.c.

7.154.3.4 `ng_constructor_t ng_tcpmss_constructor [static]`

Definition at line 72 of file ng_tcpmss.c.

7.154.3.5 `ng_disconnect_t ng_tcpmss_disconnect [static]`

Definition at line 76 of file ng_tcpmss.c.

7.154.3.6 `struct ng_parse_type ng_tcpmss_hookstat_type [static]`

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_tcpmss_hookstat_type_fields
}
```

Definition at line 83 of file ng_tcpmss.c.

7.154.3.7 `struct ng_parse_struct_field ng_tcpmss_hookstat_type_fields[] =`
`NG_TCPMSS_HOOKSTAT_INFO [static]`

Definition at line 82 of file ng_tcpmss.c.

7.154.3.8 `ng_newhook_t ng_tcpmss_newhook [static]`

Definition at line 74 of file ng_tcpmss.c.

7.154.3.9 `ng_rcvdata_t ng_tcpmss_rcvdata [static]`

Definition at line 75 of file ng_tcpmss.c.

7.154.3.10 `ng_rcvmsg_t ng_tcpmss_rcvmsg [static]`

Definition at line 73 of file ng_tcpmss.c.

7.154.3.11 `struct ng_type ng_tcpmss_tpestruct [static]`

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_TCPMSS_NODE_TYPE,
    .constructor = ng_tcpmss_constructor,
```

```
.rcvmsg =      ng_tcpmss_rcvmsg,  
.newhook =    ng_tcpmss_newhook,  
.rcvdata =    ng_tcpmss_rcvdata,  
.disconnect = ng_tcpmss_disconnect,  
.cmdlist =    ng_tcpmss_cmds,  
}
```

Definition at line 130 of file ng_tcpmss.c.

7.155 /usr/src/sys/netgraph/ng_tcpmss.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_tcpmss_hookstat](#)
- struct [ng_tcpmss_config](#)

Defines

- #define [NG_TCPMSS_NODE_TYPE](#) "tcpmss"
- #define [NGM_TCPMSS_COOKIE](#) 1097623478
- #define [NG_TCPMSS_HOOKSTAT_INFO](#)
- #define [NG_TCPMSS_CONFIG_INFO](#)

Enumerations

- enum { [NGM_TCPMSS_GET_STATS](#) = 1, [NGM_TCPMSS_CLR_STATS](#), [NGM_TCPMSS_GETCLR_STATS](#), [NGM_TCPMSS_CONFIG](#) }

7.155.1 Define Documentation

7.155.1.1 #define NG_TCPMSS_CONFIG_INFO

Value:

```

{
    { "inHook",      &ng_parse_hookbuf_type },    \
    { "outHook",    &ng_parse_hookbuf_type },    \
    { "maxMSS",     &ng_parse_uint16_type },     \
    { NULL }
}

```

Definition at line 67 of file ng_tcpmss.h.

7.155.1.2 #define NG_TCPMSS_HOOKSTAT_INFO

Value:

```

{
    { "Octets",      &ng_parse_uint64_type },    \
    { "Packets",    &ng_parse_uint64_type },    \
    { "maxMSS",     &ng_parse_uint16_type },     \
    { "SYNPkts",    &ng_parse_uint64_type },     \
    { "FixedPkts",  &ng_parse_uint64_type },     \
    { NULL }
}

```

Definition at line 49 of file ng_tcpmss.h.

7.155.1.3 #define NG_TCPMSS_NODE_TYPE "tcpmss"

Definition at line 36 of file ng_tcpmss.h.

7.155.1.4 #define NGM_TCPMSS_COOKIE 1097623478

Definition at line 37 of file ng_tcpmss.h.

Referenced by ng_tcpmss_rcvmsg().

7.155.2 Enumeration Type Documentation**7.155.2.1 anonymous enum**

Enumerator:

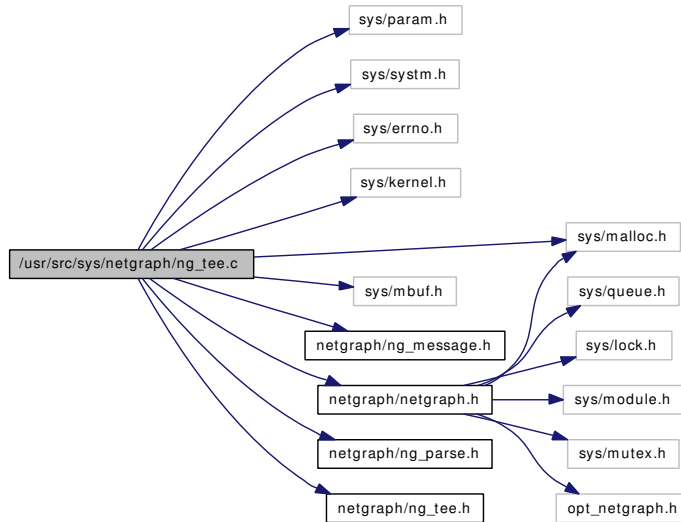
NGM_TCPMSS_GET_STATS
NGM_TCPMSS_CLR_STATS
NGM_TCPMSS_GETCLR_STATS
NGM_TCPMSS_CONFIG

Definition at line 75 of file ng_tcpmss.h.

7.156 /usr/src/sys/netgraph/ng_tee.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_tee.h>
```

Include dependency graph for ng_tee.c:



Data Structures

- struct [hookinfo](#)
- struct [privdata](#)

Typedefs

- typedef [privdata](#) * [sc_p](#)

Functions

- [NETGRAPH_INIT](#) ([tee](#), &[ng_tee_tpestruct](#))
- static int [ngt_constructor](#) ([node_p](#) node)
- static int [ngt_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *[name](#))

- static int [ngt_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ngt_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ngt_close](#) ([node_p](#) node)
- static int [ngt_shutdown](#) ([node_p](#) node)
- static int [ngt_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ngt_constructor](#)
- static [ng_rcvmsg_t](#) [ngt_rcvmsg](#)
- static [ng_close_t](#) [ngt_close](#)
- static [ng_shutdown_t](#) [ngt_shutdown](#)
- static [ng_newhook_t](#) [ngt_newhook](#)
- static [ng_rcvdata_t](#) [ngt_rcvdata](#)
- static [ng_disconnect_t](#) [ngt_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_tee_hookstat_type_fields](#) [] = NG_TEE_HOOKSTAT_INFO
- static struct [ng_parse_type](#) [ng_tee_hookstat_type](#)
- static struct [ng_parse_struct_field](#) [ng_tee_stats_type_fields](#) [] = NG_TEE_STATS_INFO(&[ng_tee_hookstat_type](#))
- static struct [ng_parse_type](#) [ng_tee_stats_type](#)
- static struct [ng_cmdlist](#) [ng_tee_cmds](#) []
- static struct [ng_type](#) [ng_tee_typestruct](#)

7.156.1 Typedef Documentation

7.156.1.1 typedef struct [privdata](#)* [sc_p](#)

Definition at line 79 of file [ng_tee.c](#).

7.156.2 Function Documentation

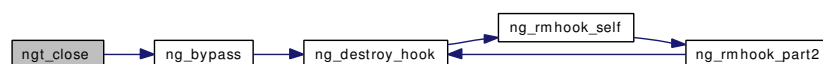
7.156.2.1 NETGRAPH_INIT ([tee](#), & [ng_tee_typestruct](#))

7.156.2.2 static int [ngt_close](#) ([node_p](#) [node](#)) [static]

Definition at line 363 of file [ng_tee.c](#).

References [ng_bypass\(\)](#), and [NG_NODE_PRIVATE](#).

Here is the call graph for this function:



7.156.2.3 static int [ngt_constructor](#) ([node_p](#) [node](#)) [static]

Definition at line 151 of file [ng_tee.c](#).

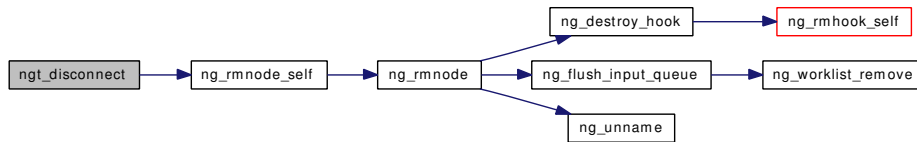
References [NG_NODE_SET_PRIVATE](#).

7.156.2.4 static int `ngt_disconnect` (`hook_p hook`) [static]

Definition at line 391 of file `ng_tee.c`.

References `hookinfo::hook`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_IS_VALID`, `NG_NODE_NUMHOOKS`, and `ng_rmnode_self()`.

Here is the call graph for this function:



7.156.2.5 static int `ngt_newhook` (`node_p node`, `hook_p hook`, `const char * name`) [static]

Definition at line 168 of file `ng_tee.c`.

References `NG_HOOK_SET_PRIVATE`, `NG_NODE_PRIVATE`, `NG_TEE_HOOK_LEFT`, `NG_TEE_HOOK_LEFT2RIGHT`, `NG_TEE_HOOK_RIGHT`, `NG_TEE_HOOK_RIGHT2LEFT`, and `ng_async_private::stats`.

7.156.2.6 static int `ngt_rcvdata` (`hook_p hook`, `item_p item`) [static]

Definition at line 288 of file `ng_tee.c`.

References `hookinfo::hook`, `ng_tee_hookstat::inFrames`, `ng_tee_hookstat::inOctets`, `NG_FREE_ITEM`, `NG_FWD_ITEM_HOOK`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `NG_SEND_DATA_ONLY`, `NGI_M`, `ng_tee_hookstat::outFrames`, `ng_tee_hookstat::outOctets`, and `hookinfo::stats`.

7.156.2.7 static int `ngt_rcvmsg` (`node_p node`, `item_p item`, `hook_p lasthook`) [static]

Definition at line 197 of file `ng_tee.c`.

References `ng_msg::ng_msghdr::cmd`, `ng_msg::header`, `NG_FREE_MSG`, `NG_FWD_ITEM_HOOK`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGI_MSG`, `NGM_FLOW_COOKIE`, `NGM_TEE_CLR_STATS`, `NGM_TEE_COOKIE`, `NGM_TEE_GET_STATS`, `NGM_TEE_GETCLR_STATS`, `ng_async_private::stats`, and `ng_msg::ng_msghdr::typecookie`.

7.156.2.8 static int `ngt_shutdown` (`node_p node`) [static]

Definition at line 377 of file `ng_tee.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, and `ng_async_private::node`.

7.156.3 Variable Documentation

7.156.3.1 struct [ng_cmdlist ng_tee_cmds\[\]](#) [static]

Initial value:

```
{
    {
        NGM_TEE_COOKIE,
        NGM_TEE_GET_STATS,
        "getstats",
        NULL,
        &ng_tee_stats_type
    },
    {
        NGM_TEE_COOKIE,
        NGM_TEE_CLR_STATS,
        "clrstats",
        NULL,
        NULL
    },
    {
        NGM_TEE_COOKIE,
        NGM_TEE_GETCLR_STATS,
        "getclrstats",
        NULL,
        &ng_tee_stats_type
    },
    { 0 }
}
```

Definition at line 107 of file ng_tee.c.

7.156.3.2 struct [ng_parse_type ng_tee_hookstat_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_tee_hookstat_type_fields
}
```

Definition at line 93 of file ng_tee.c.

7.156.3.3 struct [ng_parse_struct_field ng_tee_hookstat_type_fields\[\]](#) = [NG_TEE_HOOKSTAT_INFO](#) [static]

Definition at line 92 of file ng_tee.c.

7.156.3.4 struct [ng_parse_type ng_tee_stats_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_tee_stats_type_fields
}
```

Definition at line 101 of file ng_tee.c.

7.156.3.5 `struct ng_parse_struct_field ng_tee_stats_type_fields[] = NG_TEE_STATS_-
INFO(&ng_tee_hookstat_type)` [static]

Definition at line 100 of file ng_tee.c.

7.156.3.6 `struct ng_type ng_tee_typedstruct` [static]

Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_TEE_NODE_TYPE,
    .constructor = ngt_constructor,
    .rcvmsg =      ngt_rcvmsg,
    .close =       ngt_close,
    .shutdown =    ngt_shutdown,
    .newhook =     ngt_newhook,
    .rcvdata =     ngt_rcvdata,
    .disconnect =  ngt_disconnect,
    .cmdlist =     ng_tee_cmds,
}
```

Definition at line 133 of file ng_tee.c.

7.156.3.7 `ng_close_t ngt_close` [static]

Definition at line 84 of file ng_tee.c.

7.156.3.8 `ng_constructor_t ngt_constructor` [static]

Definition at line 82 of file ng_tee.c.

7.156.3.9 `ng_disconnect_t ngt_disconnect` [static]

Definition at line 88 of file ng_tee.c.

7.156.3.10 `ng_newhook_t ngt_newhook` [static]

Definition at line 86 of file ng_tee.c.

7.156.3.11 `ng_rcvdata_t ngt_rcvdata` [static]

Definition at line 87 of file ng_tee.c.

7.156.3.12 `ng_rcvmsg_t ngt_rcvmsg` [static]

Definition at line 83 of file ng_tee.c.

7.156.3.13 `ng_shutdown_t ngt_shutdown` [static]

Definition at line 85 of file ng_tee.c.

7.157 /usr/src/sys/netgraph/ng_tee.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_tee_hookstat](#)
- struct [ng_tee_stats](#)

Defines

- #define [NG_TEE_NODE_TYPE](#) "tee"
- #define [NGM_TEE_COOKIE](#) 916107047
- #define [NG_TEE_HOOK_RIGHT](#) "right"
- #define [NG_TEE_HOOK_LEFT](#) "left"
- #define [NG_TEE_HOOK_RIGHT2LEFT](#) "right2left"
- #define [NG_TEE_HOOK_LEFT2RIGHT](#) "left2right"
- #define [NG_TEE_HOOKSTAT_INFO](#)
- #define [NG_TEE_STATS_INFO](#)(hstype)

Enumerations

- enum { [NGM_TEE_GET_STATS](#) = 1, [NGM_TEE_CLR_STATS](#), [NGM_TEE_GETCLR_STATS](#) }

7.157.1 Define Documentation

7.157.1.1 #define [NG_TEE_HOOK_LEFT](#) "left"

Definition at line 54 of file [ng_tee.h](#).

Referenced by [ngt_newhook\(\)](#).

7.157.1.2 #define [NG_TEE_HOOK_LEFT2RIGHT](#) "left2right"

Definition at line 56 of file [ng_tee.h](#).

Referenced by [ngt_newhook\(\)](#).

7.157.1.3 #define [NG_TEE_HOOK_RIGHT](#) "right"

Definition at line 53 of file [ng_tee.h](#).

Referenced by [ngt_newhook\(\)](#).

7.157.1.4 #define NG_TEE_HOOK_RIGHT2LEFT "right2left"

Definition at line 55 of file ng_tee.h.

Referenced by ngt_newhook().

7.157.1.5 #define NG_TEE_HOOKSTAT_INFO**Value:**

```
{
    { "inOctets",      \
      &ng_parse_uint64_type }, \
    { "inFrames",     \
      &ng_parse_uint64_type }, \
    { "outOctets",    \
      &ng_parse_uint64_type }, \
    { "outFrames",    \
      &ng_parse_uint64_type }, \
    { NULL }
}
```

Definition at line 67 of file ng_tee.h.

7.157.1.6 #define NG_TEE_NODE_TYPE "tee"

Definition at line 49 of file ng_tee.h.

7.157.1.7 #define NG_TEE_STATS_INFO(hstype)**Value:**

```
{
    { "right",      \
      (hstype) }, \
    { "left",      \
      (hstype) }, \
    { "right2left", \
      (hstype) }, \
    { "left2right", \
      (hstype) }, \
    { NULL }
}
```

Definition at line 84 of file ng_tee.h.

7.157.1.8 #define NGM_TEE_COOKIE 916107047

Definition at line 50 of file ng_tee.h.

Referenced by ngt_rcvmsg().

7.157.2 Enumeration Type Documentation**7.157.2.1 anonymous enum****Enumerator:**

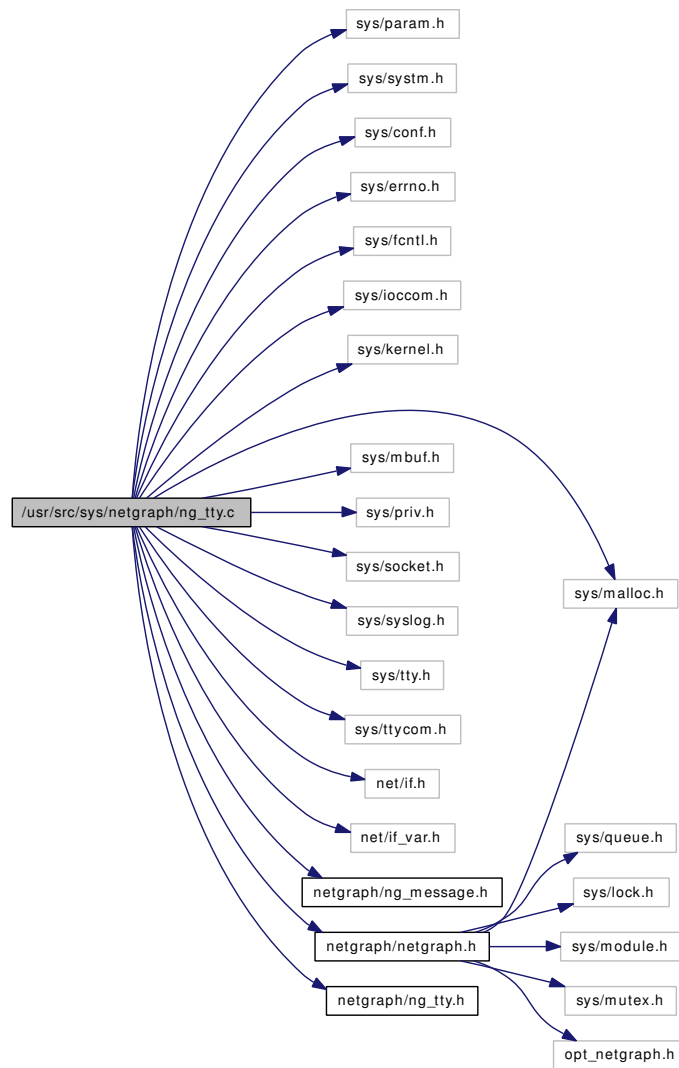
NGM_TEE_GET_STATS
NGM_TEE_CLR_STATS
NGM_TEE_GETCLR_STATS

Definition at line 93 of file ng_tee.h.

7.158 /usr/src/sys/netgraph/ng_tty.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/conf.h>
#include <sys/errno.h>
#include <sys/fcntl.h>
#include <sys/ioccom.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/priv.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/tty.h>
#include <sys/ttycom.h>
#include <net/if.h>
#include <net/if_var.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_tty.h>
```

Include dependency graph for ng_tty.c:



Data Structures

- struct [ngt_sc](#)

Defines

- #define [MAX_MBUFQ](#) 3
- #define [NGT_HIWATER](#) 400
- #define [FLG_DEBUG](#) 0x0002
- #define [FLG_DIE](#) 0x0004
- #define [ERRROUT](#)(x) do { error = (x); goto done; } while (0)
- #define [NGTLOCK](#)(sc) IF_LOCK(&sc → outq)
- #define [NGTUNLOCK](#)(sc) IF_UNLOCK(&sc → outq)

Typedefs

- typedef [ngt_sc](#) * [sc_p](#)

Functions

- static int [ngt_open](#) (struct cdev *dev, struct tty *tp)
- static int [ngt_close](#) (struct tty *tp, int flag)
- static int [ngt_read](#) (struct tty *tp, struct uio *uio, int flag)
- static int [ngt_write](#) (struct tty *tp, struct uio *uio, int flag)
- static int [ngt_tioctl](#) (struct tty *tp, u_long cmd, caddr_t data, int flag, struct thread *)
- static int [ngt_input](#) (int c, struct tty *tp)
- static int [ngt_start](#) (struct tty *tp)
- static int [ngt_mod_event](#) (module_t mod, int event, void *data)
- static void [ngt_timeout](#) ([node_p](#) node, [hook_p](#) hook, void *arg1, int arg2)
- [NETGRAPH_INIT](#) (tty,&typestruct)
- static int [ngt_constructor](#) ([node_p](#) node)
- static int [ngt_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ngt_connect](#) ([hook_p](#) hook)
- static int [ngt_disconnect](#) ([hook_p](#) hook)
- static int [ngt_shutdown](#) ([node_p](#) node)
- static int [ngt_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ngt_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)

Variables

- static [ng_constructor_t](#) [ngt_constructor](#)
- static [ng_rcvmsg_t](#) [ngt_rcvmsg](#)
- static [ng_shutdown_t](#) [ngt_shutdown](#)
- static [ng_newhook_t](#) [ngt_newhook](#)
- static [ng_connect_t](#) [ngt_connect](#)
- static [ng_rcvdata_t](#) [ngt_rcvdata](#)
- static [ng_disconnect_t](#) [ngt_disconnect](#)
- static struct linesw [ngt_disc](#)
- static struct [ng_type](#) [typestruct](#)
- static int [ngt_unit](#)
- static int [ngt_ldisc](#)

7.158.1 Define Documentation

7.158.1.1 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 126 of file [ng_tty.c](#).

7.158.1.2 #define FLG_DEBUG 0x0002

Definition at line 100 of file [ng_tty.c](#).

Referenced by [ngt_input\(\)](#).

7.158.1.3 #define FLG_DIE 0x0004

Definition at line 101 of file ng_tty.c.

Referenced by `ngt_close()`, `ngt_open()`, and `ngt_shutdown()`.

7.158.1.4 #define MAX_MBUFQ 3

Definition at line 83 of file ng_tty.c.

Referenced by `ngt_open()`.

7.158.1.5 #define NGT_HIWATER 400

Definition at line 84 of file ng_tty.c.

Referenced by `ngt_open()`, and `ngt_start()`.

7.158.1.6 #define NGTLOCK(sc) IF_LOCK(&sc → outq)

Definition at line 170 of file ng_tty.c.

Referenced by `ngt_close()`, `ngt_disconnect()`, `ngt_input()`, `ngt_newhook()`, `ngt_open()`, `ngt_shutdown()`, and `ngt_tioctl()`.

7.158.1.7 #define NGTUNLOCK(sc) IF_UNLOCK(&sc → outq)

Definition at line 171 of file ng_tty.c.

Referenced by `ngt_close()`, `ngt_disconnect()`, `ngt_input()`, `ngt_newhook()`, `ngt_open()`, `ngt_shutdown()`, and `ngt_tioctl()`.

7.158.2 Typedef Documentation**7.158.2.1 typedef struct [ngt_sc](#)* [sc_p](#)**

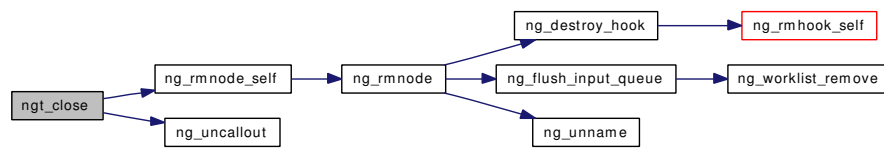
Definition at line 97 of file ng_tty.c.

7.158.3 Function Documentation**7.158.3.1 NETGRAPH_INIT (tty, & typestruct)****7.158.3.2 static int [ngt_close](#) (struct tty * *tp*, int *flag*) [static]**

Definition at line 255 of file ng_tty.c.

References `FLG_DIE`, `ng_rmnode_self()`, `ng_uncallout()`, `NGTLOCK`, and `NGTUNLOCK`.

Here is the call graph for this function:



7.158.3.3 static int `ngt_connect` (*hook_p hook*) [static]

Definition at line 517 of file `ng_tty.c`.

References `NG_HOOK_FORCE_QUEUE`, and `NG_HOOK_PEER`.

7.158.3.4 static int `ngt_constructor` (*node_p node*) [static]

Definition at line 485 of file `ng_tty.c`.

7.158.3.5 static int `ngt_disconnect` (*hook_p hook*) [static]

Definition at line 533 of file `ng_tty.c`.

References `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGTLOCK`, and `NGTUNLOCK`.

7.158.3.6 static int `ngt_input` (*int c, struct tty *tp*) [static]

Definition at line 334 of file `ng_tty.c`.

References `FLG_DEBUG`, `NG_NODE_NAME`, `NG_SEND_DATA_ONLY`, `NGTLOCK`, `NGTUNLOCK`, and `ng_async_private::node`.

7.158.3.7 static int `ngt_mod_event` (*module_t mod, int event, void *data*) [static]

Definition at line 671 of file `ng_tty.c`.

References `ngt_disc`, and `ngt_ldisc`.

7.158.3.8 static int `ngt_newhook` (*node_p node, hook_p hook, const char *name*) [static]

Definition at line 494 of file `ng_tty.c`.

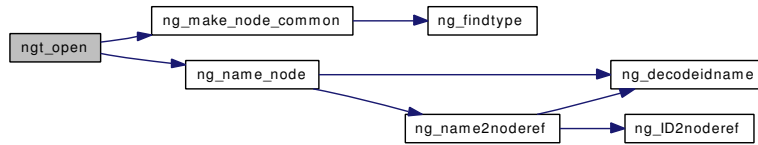
References `NG_NODE_PRIVATE`, `NG_TTY_HOOK`, `NGTLOCK`, and `NGTUNLOCK`.

7.158.3.9 static int `ngt_open` (*struct cdev *dev, struct tty *tp*) [static]

Definition at line 185 of file `ng_tty.c`.

References `FLG_DIE`, `MAX_MBUFQ`, `ng_type::name`, `name`, `ng_callout_init`, `ng_make_node_common()`, `ng_name_node()`, `NG_NODE_SET_PRIVATE`, `NG_NODE_UNREF`, `NG_TTY_DFL_HOTCHAR`, `NG_TTY_NODE_TYPE`, `NGT_HIWATER`, `ngt_unit`, `NGTLOCK`, `NGTUNLOCK`, and `typestruct`.

Here is the call graph for this function:



7.158.3.10 static int `ngt_rcvdata` (`hook_p hook`, `item_p item`) [static]

Definition at line 581 of file `ng_tty.c`.

References `NG_FREE_ITEM`, `NG_FREE_M`, `NG_HOOK_NODE`, `NG_NODE_PRIVATE`, `NGI_GET_M`, and `ngt_start()`.

Here is the call graph for this function:



7.158.3.11 static int `ngt_rcvmsg` (`node_p node`, `item_p item`, `hook_p lasthook`) [static]

Definition at line 621 of file `ng_tty.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `ERROUT`, `ng_mesg::header`, `NG_FREE_MSG`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_RESPOND_MSG`, `NGI_GET_MSG`, `NGM_TTY_COOKIE`, `NGM_TTY_GET_HOTCHAR`, `NGM_TTY_SET_HOTCHAR`, and `ng_mesg::ng_msghdr::typecookie`.

7.158.3.12 static int `ngt_read` (`struct tty * tp`, `struct uio * uio`, `int flag`) [static]

Definition at line 277 of file `ng_tty.c`.

7.158.3.13 static int `ngt_shutdown` (`node_p node`) [static]

Definition at line 555 of file `ng_tty.c`.

References `FLG_DIE`, `NG_NODE_PRIVATE`, `NG_NODE_UNREF`, `NGTLOCK`, `NGTUNLOCK`, and `ng_async_private::node`.

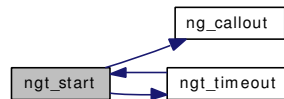
7.158.3.14 static int `ngt_start` (`struct tty * tp`) [static]

Definition at line 416 of file `ng_tty.c`.

References `ng_callout()`, `NGT_HIWATER`, `ngt_timeout()`, and `ng_async_private::node`.

Referenced by `ngt_rcvdata()`, and `ngt_timeout()`.

Here is the call graph for this function:



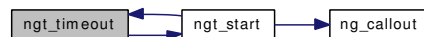
7.158.3.15 `static void ng_timeout (node_p node, hook_p hook, void * arg1, int arg2)` [static]

Definition at line 465 of file ng_tty.c.

References NG_NODE_PRIVATE, and ng_start().

Referenced by ng_start().

Here is the call graph for this function:



7.158.3.16 `static int ng_tioctl (struct tty * tp, u_long cmd, caddr_t data, int flag, struct thread *)` [static]

Definition at line 295 of file ng_tty.c.

References ng_type::name, ng_node::nd_type, ng_node2ID(), NG_NODE_HAS_NAME, NG_NODE_NAME, NG_NODE_NUMHOOKS, NGIOCGINFO, NGTLOCK, NGTUNLOCK, and ng_async_private::node.

Here is the call graph for this function:



7.158.3.17 `static int ng_write (struct tty * tp, struct uio * uio, int flag)` [static]

Definition at line 286 of file ng_tty.c.

7.158.4 Variable Documentation

7.158.4.1 `ng_connect_t ng_connect` [static]

Definition at line 118 of file ng_tty.c.

7.158.4.2 `ng_constructor_t ng_constructor` [static]

Definition at line 114 of file ng_tty.c.

7.158.4.3 struct linesw [ngt_disc](#) [static]**Initial value:**

```
{
    .l_open =      ngt_open,
    .l_close =    ngt_close,
    .l_read =     ngt_read,
    .l_write =    ngt_write,
    .l_ioctl =    ngt_tioctl,
    .l_rint =     ngt_input,
    .l_start =    ngt_start,
    .l_modem =    ttymodem,
}
```

Definition at line 129 of file ng_tty.c.

Referenced by ngt_mod_event().

7.158.4.4 [ng_disconnect_t](#) [ngt_disconnect](#) [static]

Definition at line 120 of file ng_tty.c.

7.158.4.5 int [ngt_ldisc](#) [static]

Definition at line 174 of file ng_tty.c.

Referenced by ngt_mod_event().

7.158.4.6 [ng_newhook_t](#) [ngt_newhook](#) [static]

Definition at line 117 of file ng_tty.c.

7.158.4.7 [ng_rcvdata_t](#) [ngt_rcvdata](#) [static]

Definition at line 119 of file ng_tty.c.

7.158.4.8 [ng_rcvmsg_t](#) [ngt_rcvmsg](#) [static]

Definition at line 115 of file ng_tty.c.

7.158.4.9 [ng_shutdown_t](#) [ngt_shutdown](#) [static]

Definition at line 116 of file ng_tty.c.

7.158.4.10 int [ngt_unit](#) [static]

Definition at line 173 of file ng_tty.c.

Referenced by ngt_open().

7.158.4.11 struct ng_type typestruct [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_TTY_NODE_TYPE,
    .mod_event =   ngt_mod_event,
    .constructor = ngt_constructor,
    .rcvmsg =      ngt_rcvmsg,
    .shutdown =   ngt_shutdown,
    .newhook =    ngt_newhook,
    .connect =    ngt_connect,
    .rcvdata =    ngt_rcvdata,
    .disconnect = ngt_disconnect,
}
```

Definition at line 141 of file ng_tty.c.

7.159 /usr/src/sys/netgraph/ng_tty.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [NG_TTY_NODE_TYPE](#) "tty"
- #define [NGM_TTY_COOKIE](#) 886279262
- #define [NG_TTY_DFL_HOTCHAR](#) 0x7e
- #define [NG_TTY_HOOK](#) "hook"

Enumerations

- enum { [NGM_TTY_GET_HOTCHAR](#) = 1, [NGM_TTY_SET_HOTCHAR](#) }

7.159.1 Define Documentation

7.159.1.1 #define [NG_TTY_DFL_HOTCHAR](#) 0x7e

Definition at line 53 of file ng_tty.h.

Referenced by `ngt_open()`.

7.159.1.2 #define [NG_TTY_HOOK](#) "hook"

Definition at line 56 of file ng_tty.h.

Referenced by `ngt_newhook()`.

7.159.1.3 #define [NG_TTY_NODE_TYPE](#) "tty"

Definition at line 49 of file ng_tty.h.

Referenced by `ngt_open()`.

7.159.1.4 #define [NGM_TTY_COOKIE](#) 886279262

Definition at line 50 of file ng_tty.h.

Referenced by `ngt_rcvmsg()`.

7.159.2 Enumeration Type Documentation

7.159.2.1 anonymous enum

Enumerator:

[NGM_TTY_GET_HOTCHAR](#)

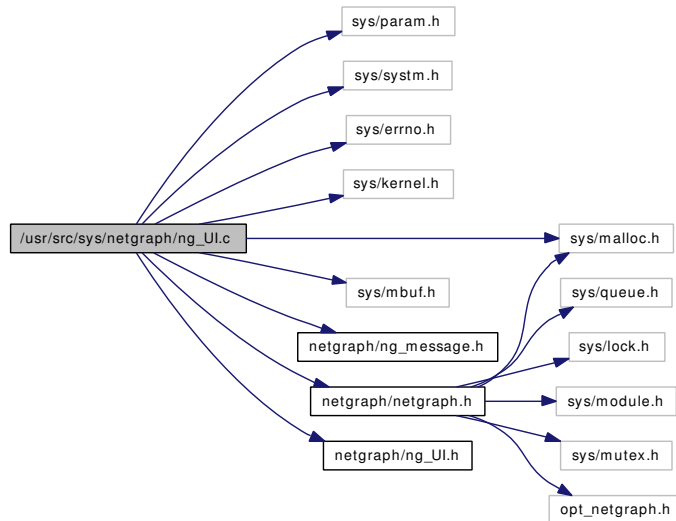
NGM_TTY_SET_HOTCHAR

Definition at line 59 of file ng_tty.h.

7.160 /usr/src/sys/netgraph/ng_UI.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_UI.h>
```

Include dependency graph for ng_UI.c:



Data Structures

- struct [ng_UI_private](#)

Defines

- #define [HDLC_UI](#) 0x03
- #define [MAX_ENCAPS_HDR](#) 1
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)

Typedefs

- typedef [ng_UI_private](#) * [priv_p](#)

Functions

- `NETGRAPH_INIT` (UI,&typestruct)
- static int `ng_UI_constructor` (node_p node)
- static int `ng_UI_newhook` (node_p node, hook_p hook, const char *name)
- static int `ng_UI_rcvmsg` (node_p node, item_p item, hook_p lasthook)
- static int `ng_UI_rcvdata` (hook_p hook, item_p item)
- static int `ng_UI_shutdown` (node_p node)
- static int `ng_UI_disconnect` (hook_p hook)

Variables

- static `ng_constructor_t` `ng_UI_constructor`
- static `ng_rcvmsg_t` `ng_UI_rcvmsg`
- static `ng_shutdown_t` `ng_UI_shutdown`
- static `ng_newhook_t` `ng_UI_newhook`
- static `ng_rcvdata_t` `ng_UI_rcvdata`
- static `ng_disconnect_t` `ng_UI_disconnect`
- static struct `ng_type` `typestruct`

7.160.1 Define Documentation

7.160.1.1 `#define ERROUT(x) do { error = (x); goto done; } while (0)`

Definition at line 165 of file `ng_UI.c`.

7.160.1.2 `#define HDLC_UI 0x03`

Definition at line 62 of file `ng_UI.c`.

7.160.1.3 `#define MAX_ENCAPS_HDR 1`

Definition at line 164 of file `ng_UI.c`.

7.160.2 Typedef Documentation

7.160.2.1 `typedef struct ng_UI_private* priv_p`

Definition at line 69 of file `ng_UI.c`.

7.160.3 Function Documentation

7.160.3.1 `NETGRAPH_INIT` (UI, & *typestruct*)

7.160.3.2 `static int ng_UI_constructor` (node_p *node*) [static]

Definition at line 101 of file `ng_UI.c`.

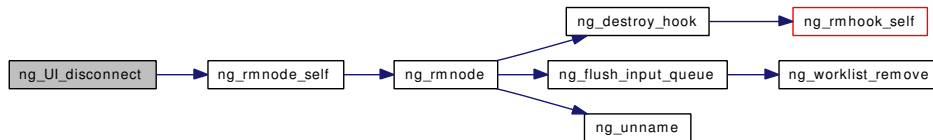
References `NG_NODE_SET_PRIVATE`.

7.160.3.3 static int ng_UI_disconnect (hook_p hook) [static]

Definition at line 228 of file ng_UI.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:

**7.160.3.4 static int ng_UI_newhook (node_p node, hook_p hook, const char * name) [static]**

Definition at line 118 of file ng_UI.c.

References NG_NODE_PRIVATE, NG_UI_HOOK_DOWNSTREAM, and NG_UI_HOOK_UPSTREAM.

7.160.3.5 static int ng_UI_rcvdata (hook_p hook, item_p item) [static]

Definition at line 171 of file ng_UI.c.

References ERROUT, HDLC_UI, MAX_ENCAPS_HDR, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_NODE_PRIVATE, and NGI_GET_M.

7.160.3.6 static int ng_UI_rcvmsg (node_p node, item_p item, hook_p lasthook) [static]

Definition at line 139 of file ng_UI.c.

References ng_mesg::header, NG_FREE_ITEM, NG_FWD_ITEM_HOOK, NG_NODE_PRIVATE, NGI_MSG, NGM_FLOW_COOKIE, and ng_mesg::ng_msghdr::typecookie.

7.160.3.7 static int ng_UI_shutdown (node_p node) [static]

Definition at line 213 of file ng_UI.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.160.4 Variable Documentation**7.160.4.1 ng_constructor_t ng_UI_constructor [static]**

Definition at line 72 of file ng_UI.c.

7.160.4.2 ng_disconnect_t ng_UI_disconnect [static]

Definition at line 77 of file ng_UI.c.

7.160.4.3 `ng_newhook_t ng_UI_newhook` [static]

Definition at line 75 of file ng_UI.c.

7.160.4.4 `ng_rcvdata_t ng_UI_rcvdata` [static]

Definition at line 76 of file ng_UI.c.

7.160.4.5 `ng_rcvmsg_t ng_UI_rcvmsg` [static]

Definition at line 73 of file ng_UI.c.

7.160.4.6 `ng_shutdown_t ng_UI_shutdown` [static]

Definition at line 74 of file ng_UI.c.

7.160.4.7 `struct ng_type typestruct` [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =         NG_UI_NODE_TYPE,
    .constructor = ng_UI_constructor,
    .rcvmsg =       ng_UI_rcvmsg,
    .shutdown =     ng_UI_shutdown,
    .newhook =      ng_UI_newhook,
    .rcvdata =      ng_UI_rcvdata,
    .disconnect =   ng_UI_disconnect,
}
```

Definition at line 80 of file ng_UI.c.

7.161 /usr/src/sys/netgraph/ng_UI.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `NG_UI_NODE_TYPE` "UI"
- #define `NGM_UI_COOKIE` 884639499
- #define `NG_UI_HOOK_DOWNSTREAM` "downstream"
- #define `NG_UI_HOOK_UPSTREAM` "upstream"

7.161.1 Define Documentation

7.161.1.1 #define `NG_UI_HOOK_DOWNSTREAM` "downstream"

Definition at line 52 of file `ng_UI.h`.

Referenced by `ng_UI_newhook()`.

7.161.1.2 #define `NG_UI_HOOK_UPSTREAM` "upstream"

Definition at line 53 of file `ng_UI.h`.

Referenced by `ng_UI_newhook()`.

7.161.1.3 #define `NG_UI_NODE_TYPE` "UI"

Definition at line 48 of file `ng_UI.h`.

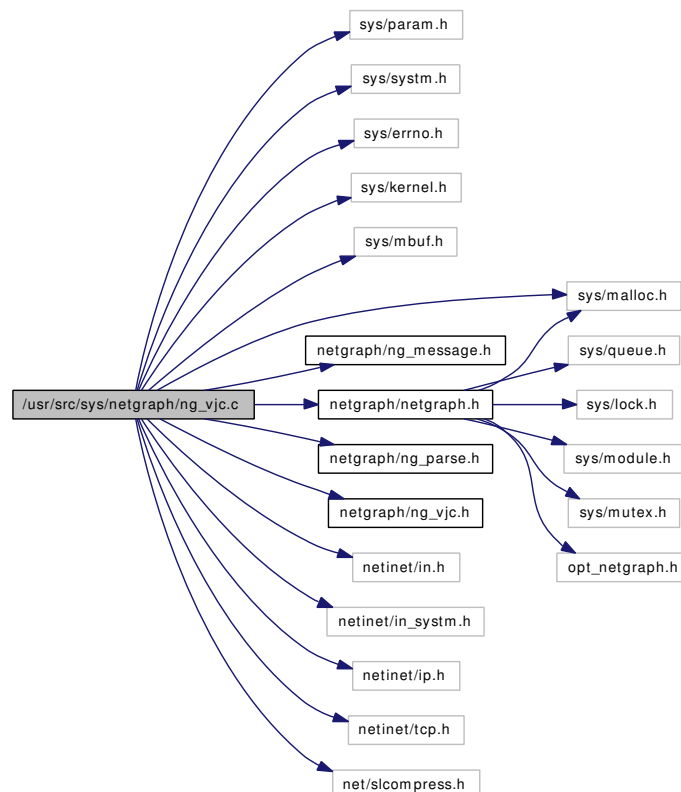
7.161.1.4 #define `NGM_UI_COOKIE` 884639499

Definition at line 49 of file `ng_UI.h`.

7.162 /usr/src/sys/netgraph/ng_vjc.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/mbuf.h>
#include <sys/malloc.h>
#include <netgraph/ng_message.h>
#include <netgraph/netgraph.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_vjc.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <net/slcompress.h>
```

Include dependency graph for ng_vjc.c:



Data Structures

- struct [ng_vjc_private](#)

Defines

- #define [MAX_VJHEADER](#) 19
- #define [ERROUT](#)(x) do { error = (x); goto done; } while (0)
- #define [NG_VJC_TSTATE_PTR_TYPE](#) [&ng_parse_uint32_type](#)

Typedefs

- typedef [ng_vjc_private](#) * [priv_p](#)

Functions

- static struct mbuf * [ng_vjc_pulluphdrs](#) (struct mbuf *m, int knownTCP)
- [NETGRAPH_INIT](#) (vjc,&ng_vjc_typestruct)
- static int [ng_vjc_constructor](#) (node_p node)
- static int [ng_vjc_newhook](#) (node_p node, hook_p hook, const char *name)
- static int [ng_vjc_rcvmsg](#) (node_p node, item_p item, hook_p lasthook)
- static int [ng_vjc_rcvdata](#) (hook_p hook, item_p item)
- static int [ng_vjc_shutdown](#) (node_p node)
- static int [ng_vjc_disconnect](#) (hook_p hook)

Variables

- static [ng_constructor_t](#) [ng_vjc_constructor](#)
- static [ng_rcvmsg_t](#) [ng_vjc_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_vjc_shutdown](#)
- static [ng_newhook_t](#) [ng_vjc_newhook](#)
- static [ng_rcvdata_t](#) [ng_vjc_rcvdata](#)
- static [ng_disconnect_t](#) [ng_vjc_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_vjc_config_type_fields](#) [] = [NG_VJC_CONFIG_TYPE_INFO](#)
- static struct [ng_parse_type](#) [ng_vjc_config_type](#)
- static struct [ng_parse_fixedarray_info](#) [ng_vjc_cs_hdr_type_info](#)
- static struct [ng_parse_type](#) [ng_vjc_cs_hdr_type](#)
- static struct [ng_parse_struct_field](#) [ng_vjc_cstate_type_fields](#) []
- static struct [ng_parse_type](#) [ng_vjc_cstate_type](#)
- static struct [ng_parse_fixedarray_info](#) [ng_vjc_cstatearray_type_info](#)
- static struct [ng_parse_type](#) [ng_vjc_cstatearray_type](#)
- static struct [ng_parse_struct_field](#) [ng_vjc_slcompress_type_fields](#) []
- static struct [ng_parse_type](#) [ng_vjc_slcompress_type](#)
- static struct [ng_cmdlist](#) [ng_vjc_cmds](#) []
- static struct [ng_type](#) [ng_vjc_typestruct](#)

7.162.1 Define Documentation

7.162.1.1 #define ERROUT(x) do { error = (x); goto done; } while (0)

Definition at line 89 of file ng_vjc.c.

7.162.1.2 #define MAX_VJHEADER 19

Definition at line 76 of file ng_vjc.c.

Referenced by ng_vjc_rcvdata().

7.162.1.3 #define NG_VJC_TSTATE_PTR_TYPE &ng_parse_uint32_type

Definition at line 113 of file ng_vjc.c.

7.162.2 Typedef Documentation

7.162.2.1 typedef struct ng_vjc_private* priv_p

Definition at line 87 of file ng_vjc.c.

7.162.3 Function Documentation

7.162.3.1 NETGRAPH_INIT (vjc, & ng_vjc_typestruct)

7.162.3.2 static int ng_vjc_constructor (node_p node) [static]

Definition at line 241 of file ng_vjc.c.

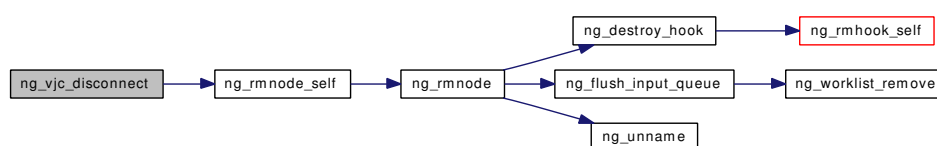
References NG_NODE_SET_PRIVATE.

7.162.3.3 static int ng_vjc_disconnect (hook_p hook) [static]

Definition at line 558 of file ng_vjc.c.

References NG_HOOK_NODE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, and ng_rmnode_self().

Here is the call graph for this function:



7.162.3.4 `static int ng_vjc_newhook (node_p node, hook_p hook, const char * name)` [static]

Definition at line 260 of file ng_vjc.c.

References NG_NODE_PRIVATE, NG_VJC_HOOK_IP, NG_VJC_HOOK_VJCOMP, NG_VJC_HOOK_VJIP, and NG_VJC_HOOK_VJUNCOMP.

7.162.3.5 `static struct mbuf * ng_vjc_pulluphdrs (struct mbuf * m, int knownTCP)` [static]

Definition at line 591 of file ng_vjc.c.

Referenced by ng_vjc_rcvdata().

7.162.3.6 `static int ng_vjc_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 397 of file ng_vjc.c.

References MAX_VJHEADER, NG_FREE_ITEM, NG_FREE_M, NG_FWD_NEW_DATA, NG_HOOK_NODE, NG_NODE_PRIVATE, ng_vjc_pulluphdrs(), and NGI_GET_M.

Here is the call graph for this function:

**7.162.3.7** `static int ng_vjc_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 290 of file ng_vjc.c.

References ng_mesg::ng_msghdr::arglen, ng_mesg::ng_msghdr::cmd, ng_mesg::data, ERROUT, ng_mesg::header, NG_FREE_MSG, NG_MKRESPONSE, NG_NODE_PRIVATE, NG_RESPOND_MSG, NG_VJC_MAX_CHANNELS, NG_VJC_MIN_CHANNELS, NGI_GET_MSG, NGM_VJC_CLR_STATS, NGM_VJC_COOKIE, NGM_VJC_GET_CONFIG, NGM_VJC_GET_STATE, NGM_VJC_RECV_ERROR, NGM_VJC_SET_CONFIG, and ng_mesg::ng_msghdr::typecookie.

7.162.3.8 `static int ng_vjc_shutdown (node_p node)` [static]

Definition at line 543 of file ng_vjc.c.

References NG_NODE_PRIVATE, NG_NODE_SET_PRIVATE, and NG_NODE_UNREF.

7.162.4 Variable Documentation**7.162.4.1** `struct ng_cmdlist ng_vjc_cmds[]` [static]

Definition at line 180 of file ng_vjc.c.

7.162.4.2 `struct ng_parse_type ng_vjc_config_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_vjc_config_type_fields
}
```

Definition at line 105 of file ng_vjc.c.

7.162.4.3 struct [ng_parse_struct_field](#) [ng_vjc_config_type_fields](#)[] =
NG_VJC_CONFIG_TYPE_INFO [static]

Definition at line 104 of file ng_vjc.c.

7.162.4.4 [ng_constructor_t](#) [ng_vjc_constructor](#) [static]

Definition at line 92 of file ng_vjc.c.

7.162.4.5 struct [ng_parse_type](#) [ng_vjc_cs_hdr_type](#) [static]

Initial value:

```
{
    &ng_parse_fixedarray_type,
    &ng_vjc_cs_hdr_type_info
}
```

Definition at line 124 of file ng_vjc.c.

7.162.4.6 struct [ng_parse_fixedarray_info](#) [ng_vjc_cs_hdr_type_info](#) [static]

Initial value:

```
{
    &ng_parse_hint8_type,
    MAX_HDR
}
```

Definition at line 120 of file ng_vjc.c.

7.162.4.7 struct [ng_parse_type](#) [ng_vjc_cstate_type](#) [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_vjc_cstate_type_fields
}
```

Definition at line 138 of file ng_vjc.c.

7.162.4.8 struct [ng_parse_struct_field ng_vjc_cstate_type_fields\[\]](#) [static]**Initial value:**

```

{
    { "cs_next",          NG_VJC_TSTATE_PTR_TYPE      },
    { "cs_hlen",         &ng_parse_uint16_type      },
    { "cs_id",           &ng_parse_uint8_type       },
    { "cs_filler",       &ng_parse_uint8_type       },
    { "cs_hdr",          &ng_vjc_cs_hdr_type        },
    { NULL }
}

```

Definition at line 130 of file ng_vjc.c.

7.162.4.9 struct [ng_parse_type ng_vjc_cstatearray_type](#) [static]**Initial value:**

```

{
    &ng_parse_fixedarray_type,
    &ng_vjc_cstatearray_type_info
}

```

Definition at line 148 of file ng_vjc.c.

7.162.4.10 struct [ng_parse_fixedarray_info ng_vjc_cstatearray_type_info](#) [static]**Initial value:**

```

{
    &ng_vjc_cstate_type,
    MAX_STATES
}

```

Definition at line 144 of file ng_vjc.c.

7.162.4.11 [ng_disconnect_t ng_vjc_disconnect](#) [static]

Definition at line 97 of file ng_vjc.c.

7.162.4.12 [ng_newhook_t ng_vjc_newhook](#) [static]

Definition at line 95 of file ng_vjc.c.

7.162.4.13 [ng_rcvdata_t ng_vjc_rcvdata](#) [static]

Definition at line 96 of file ng_vjc.c.

7.162.4.14 [ng_rcvmsg_t ng_vjc_rcvmsg](#) [static]

Definition at line 93 of file ng_vjc.c.

7.162.4.15 `ng_shutdown_t ng_vjc_shutdown` [static]

Definition at line 94 of file `ng_vjc.c`.

7.162.4.16 `struct ng_parse_type ng_vjc_slcompress_type` [static]

Initial value:

```
{
    &ng_parse_struct_type,
    &ng_vjc_slcompress_type_fields
}
```

Definition at line 174 of file `ng_vjc.c`.

7.162.4.17 `struct ng_parse_struct_field ng_vjc_slcompress_type_fields[]` [static]

Initial value:

```
{
    { "last_cs",          NG_VJC_TSTATE_PTR_TYPE      },
    { "last_rcv",        &ng_parse_uint8_type      },
    { "last_xmit",       &ng_parse_uint8_type      },
    { "flags",           &ng_parse_hint16_type     },

    { "sls_packets",     &ng_parse_uint32_type     },
    { "sls_compressed",  &ng_parse_uint32_type     },
    { "sls_searches",   &ng_parse_uint32_type     },
    { "sls_misses",     &ng_parse_uint32_type     },
    { "sls_uncompressedin", &ng_parse_uint32_type    },
    { "sls_compressedin", &ng_parse_uint32_type    },
    { "sls_errorin",    &ng_parse_uint32_type    },
    { "sls_tossed",     &ng_parse_uint32_type    },

    { "tstate",         &ng_vjc_cstatearray_type  },
    { "rstate",         &ng_vjc_cstatearray_type  },
    { NULL }
}
```

Definition at line 155 of file `ng_vjc.c`.

7.162.4.18 `struct ng_type ng_vjc_typestruct` [static]

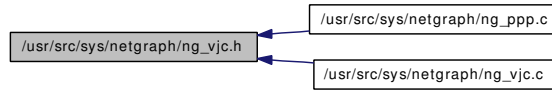
Initial value:

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_VJC_NODE_TYPE,
    .constructor = ng_vjc_constructor,
    .rcvmsg =      ng_vjc_rcvmsg,
    .shutdown =    ng_vjc_shutdown,
    .newhook =     ng_vjc_newhook,
    .rcvdata =     ng_vjc_rcvdata,
    .disconnect =  ng_vjc_disconnect,
    .cmdlist =     ng_vjc_cmds,
}
```

Definition at line 220 of file `ng_vjc.c`.

7.163 /usr/src/sys/netgraph/ng_vjc.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ngm_vjc_config](#)

Defines

- #define [NG_VJC_NODE_TYPE](#) "vjc"
- #define [NGM_VJC_COOKIE](#) 868219210
- #define [NG_VJC_HOOK_IP](#) "ip"
- #define [NG_VJC_HOOK_VJCOMP](#) "vjcomp"
- #define [NG_VJC_HOOK_VJUNCOMP](#) "vjuncomp"
- #define [NG_VJC_HOOK_VJIP](#) "vjip"
- #define [NG_VJC_MIN_CHANNELS](#) 4
- #define [NG_VJC_MAX_CHANNELS](#) 16
- #define [NG_VJC_CONFIG_TYPE_INFO](#)

Enumerations

- enum {
 - [NGM_VJC_SET_CONFIG](#), [NGM_VJC_GET_CONFIG](#), [NGM_VJC_GET_STATE](#), [NGM_VJC_CLR_STATS](#),
 - [NGM_VJC_RECV_ERROR](#) }

7.163.1 Define Documentation

7.163.1.1 #define NG_VJC_CONFIG_TYPE_INFO

Value:

```

{
    { "enableComp",      \
      &ng_parse_uint8_type }, \
    { "enableDecomp",   \
      &ng_parse_uint8_type }, \
    { "maxChannel",     \
      &ng_parse_uint8_type }, \
    { "compressCID",    \
      &ng_parse_uint8_type }, \
    { NULL }
}

```

Definition at line 71 of file [ng_vjc.h](#).

7.163.1.2 #define NG_VJC_HOOK_IP "ip"

Definition at line 53 of file ng_vjc.h.

Referenced by ng_vjc_newhook().

7.163.1.3 #define NG_VJC_HOOK_VJCOMP "vjcomp"

Definition at line 54 of file ng_vjc.h.

Referenced by ng_vjc_newhook().

7.163.1.4 #define NG_VJC_HOOK_VJIP "vjip"

Definition at line 56 of file ng_vjc.h.

Referenced by ng_vjc_newhook().

7.163.1.5 #define NG_VJC_HOOK_VJUNCOMP "vjuncomp"

Definition at line 55 of file ng_vjc.h.

Referenced by ng_vjc_newhook().

7.163.1.6 #define NG_VJC_MAX_CHANNELS 16

Definition at line 60 of file ng_vjc.h.

Referenced by ng_vjc_rcvmsg().

7.163.1.7 #define NG_VJC_MIN_CHANNELS 4

Definition at line 59 of file ng_vjc.h.

Referenced by ng_vjc_rcvmsg().

7.163.1.8 #define NG_VJC_NODE_TYPE "vjc"

Definition at line 49 of file ng_vjc.h.

7.163.1.9 #define NGM_VJC_COOKIE 868219210

Definition at line 50 of file ng_vjc.h.

Referenced by ng_ppp_rcvmsg(), and ng_vjc_rcvmsg().

7.163.2 Enumeration Type Documentation

7.163.2.1 anonymous enum

Enumerator:

NGM_VJC_SET_CONFIG

NGM_VJC_GET_CONFIG

NGM_VJC_GET_STATE

NGM_VJC_CLR_STATS

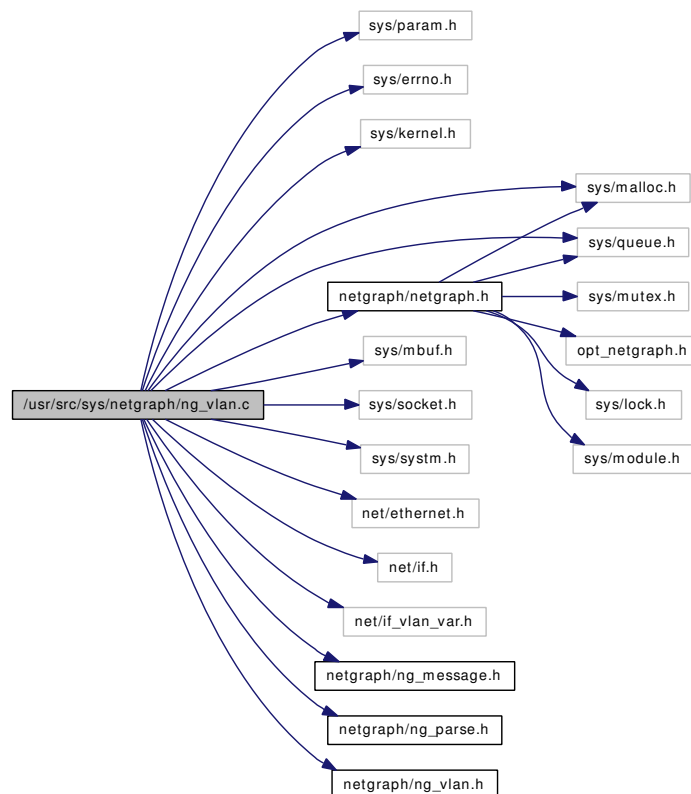
NGM_VJC_RECV_ERROR

Definition at line 80 of file ng_vjc.h.

7.164 /usr/src/sys/netgraph/ng_vlan.c File Reference

```
#include <sys/param.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/system.h>
#include <net/ethernet.h>
#include <net/if.h>
#include <net/if_vlan_var.h>
#include <netgraph/ng_message.h>
#include <netgraph/ng_parse.h>
#include <netgraph/ng_vlan.h>
#include <netgraph/netgraph.h>
```

Include dependency graph for ng_vlan.c:



Data Structures

- struct [filter](#)
- struct [priv_p](#)

Defines

- #define [HASHSIZE](#) 16
- #define [HASH\(id\)](#) (((id) >> 8) ^ ((id) >> 4) ^ (id)) & 0x0f)

Functions

- static int [ng_vlan_getTableLength](#) (const struct [ng_parse_type](#) *type, const u_char *start, const u_char *buf)
- [NETGRAPH_INIT](#) (vlan,&[ng_vlan_tpestruct](#))
- [LIST_HEAD](#) (filterhead, [filter](#))
- static struct [filter](#) * [ng_vlan_findentry](#) ([priv_p](#) priv, u_int16_t vlan)
- static int [ng_vlan_constructor](#) ([node_p](#) node)
- static int [ng_vlan_newhook](#) ([node_p](#) node, [hook_p](#) hook, const char *name)
- static int [ng_vlan_rcvmsg](#) ([node_p](#) node, [item_p](#) item, [hook_p](#) lasthook)
- static int [ng_vlan_rcvdata](#) ([hook_p](#) hook, [item_p](#) item)
- static int [ng_vlan_shutdown](#) ([node_p](#) node)
- static int [ng_vlan_disconnect](#) ([hook_p](#) hook)

Variables

- static [ng_constructor_t](#) [ng_vlan_constructor](#)
- static [ng_rcvmsg_t](#) [ng_vlan_rcvmsg](#)
- static [ng_shutdown_t](#) [ng_vlan_shutdown](#)
- static [ng_newhook_t](#) [ng_vlan_newhook](#)
- static [ng_rcvdata_t](#) [ng_vlan_rcvdata](#)
- static [ng_disconnect_t](#) [ng_vlan_disconnect](#)
- static struct [ng_parse_struct_field](#) [ng_vlan_filter_fields](#) []
- static struct [ng_parse_type](#) [ng_vlan_filter_type](#)
- static struct [ng_parse_array_info](#) [ng_vlan_table_array_info](#)
- static struct [ng_parse_type](#) [ng_vlan_table_array_type](#)
- static struct [ng_parse_struct_field](#) [ng_vlan_table_fields](#) []
- static struct [ng_parse_type](#) [ng_vlan_table_type](#)
- static struct [ng_cmdlist](#) [ng_vlan_cmdlist](#) []
- static struct [ng_type](#) [ng_vlan_tpestruct](#)

7.164.1 Define Documentation

7.164.1.1 #define [HASH\(id\)](#) (((id) >> 8) ^ ((id) >> 4) ^ (id)) & 0x0f)

Definition at line 136 of file [ng_vlan.c](#).

7.164.1.2 #define HASHSIZE 16

Definition at line 135 of file ng_vlan.c.

7.164.2 Function Documentation**7.164.2.1 LIST_HEAD (filterhead, filter)****7.164.2.2 NETGRAPH_INIT (vlan, & ng_vlan_tpestruct)****7.164.2.3 static int ng_vlan_constructor (node_p node) [static]**

Definition at line 159 of file ng_vlan.c.

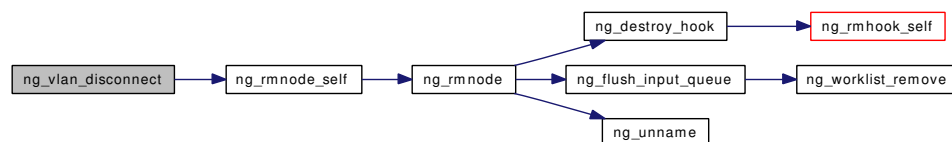
References HASHSIZE, and NG_NODE_SET_PRIVATE.

7.164.2.4 static int ng_vlan_disconnect (hook_p hook) [static]

Definition at line 447 of file ng_vlan.c.

References priv_p::downstream_hook, priv_p::nent, NG_HOOK_NODE, NG_HOOK_PRIVATE, NG_HOOK_SET_PRIVATE, NG_NODE_IS_VALID, NG_NODE_NUMHOOKS, NG_NODE_PRIVATE, ng_rmnode_self(), and priv_p::nomatch_hook.

Here is the call graph for this function:

**7.164.2.5 static struct filter* ng_vlan_findextry (priv_p priv, u_int16_t vlan) [static]**

Definition at line 147 of file ng_vlan.c.

References HASH, and priv_p::hashtable.

Referenced by ng_vlan_rcvdata(), and ng_vlan_rcvmsg().

7.164.2.6 static int ng_vlan_getTableLength (const struct ng_parse_type * type, const u_char * start, const u_char * buf) [static]

Definition at line 65 of file ng_vlan.c.

References ng_vlan_table::n.

7.164.2.7 static int ng_vlan_newhook (node_p node, hook_p hook, const char * name) [static]

Definition at line 174 of file ng_vlan.c.

References `priv_p::downstream_hook`, `NG_HOOK_SET_PRIVATE`, `NG_NODE_PRIVATE`, `NG_VLAN_HOOK_DOWNSTREAM`, `NG_VLAN_HOOK_NOMATCH`, and `priv_p::nomatch_hook`.

7.164.2.8 `static int ng_vlan_rcvdata (hook_p hook, item_p item)` [static]

Definition at line 340 of file `ng_vlan.c`.

References `priv_p::downstream_hook`, `NG_FREE_ITEM`, `NG_FREE_M`, `NG_FWD_NEW_DATA`, `NG_HOOK_NODE`, `NG_HOOK_PRIVATE`, `NG_NODE_PRIVATE`, `ng_vlan_findextry()`, `NGI_GET_M`, and `priv_p::nomatch_hook`.

Here is the call graph for this function:

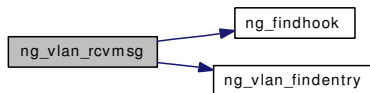


7.164.2.9 `static int ng_vlan_rcvmsg (node_p node, item_p item, hook_p lasthook)` [static]

Definition at line 193 of file `ng_vlan.c`.

References `ng_mesg::ng_msghdr::arglen`, `ng_mesg::ng_msghdr::cmd`, `ng_mesg::data`, `priv_p::downstream_hook`, `HASH`, `HASHSIZE`, `priv_p::hashtable`, `ng_mesg::header`, `ng_vlan_filter::hook`, `priv_p::nent`, `NG_COPYMESSAGE`, `ng_findexhook()`, `NG_HOOK_NAME`, `NG_HOOK_PRIVATE`, `NG_HOOK_SET_PRIVATE`, `NG_HOOKSIZ`, `NG_MKRESPONSE`, `NG_NODE_PRIVATE`, `NG_SEND_MSG_HOOK`, `ng_vlan_findextry()`, `NGI_GET_MSG`, `NGM_FLOW_COOKIE`, `NGM_VLAN_ADD_FILTER`, `NGM_VLAN_COOKIE`, `NGM_VLAN_DEL_FILTER`, `NGM_VLAN_GET_TABLE`, `priv_p::nomatch_hook`, and `ng_mesg::ng_msghdr::typecookie`.

Here is the call graph for this function:



7.164.2.10 `static int ng_vlan_shutdown (node_p node)` [static]

Definition at line 436 of file `ng_vlan.c`.

References `NG_NODE_PRIVATE`, `NG_NODE_SET_PRIVATE`, and `NG_NODE_UNREF`.

7.164.3 Variable Documentation

7.164.3.1 `struct ng_cmdlist ng_vlan_cmdlist[]` [static]

Initial value:

```

{
    {
        NGM_VLAN_COOKIE,
        NGM_VLAN_ADD_FILTER,
    }
}
  
```

```

        "addfilter",
        &ng_vlan_filter_type,
        NULL
    },
    {
        NGM_VLAN_COOKIE,
        NGM_VLAN_DEL_FILTER,
        "delfilter",
        &ng_parse_hookbuf_type,
        NULL
    },
    {
        NGM_VLAN_COOKIE,
        NGM_VLAN_GET_TABLE,
        "gettable",
        NULL,
        &ng_vlan_table_type
    },
    { 0 }
}

```

Definition at line 91 of file ng_vlan.c.

7.164.3.2 [ng_constructor_t ng_vlan_constructor](#) [static]

Definition at line 49 of file ng_vlan.c.

7.164.3.3 [ng_disconnect_t ng_vlan_disconnect](#) [static]

Definition at line 54 of file ng_vlan.c.

7.164.3.4 [struct ng_parse_struct_field ng_vlan_filter_fields\[\]](#) [static]

Initial value:

```

    NG_VLAN_FILTER_FIELDS

```

Definition at line 57 of file ng_vlan.c.

7.164.3.5 [struct ng_parse_type ng_vlan_filter_type](#) [static]

Initial value:

```

{
    &ng_parse_struct_type,
    &ng_vlan_filter_fields
}

```

Definition at line 59 of file ng_vlan.c.

7.164.3.6 [ng_newhook_t ng_vlan_newhook](#) [static]

Definition at line 52 of file ng_vlan.c.

7.164.3.7 `ng_rcvdata_t ng_vlan_rcvdata` [static]

Definition at line 53 of file `ng_vlan.c`.

7.164.3.8 `ng_rcvmsg_t ng_vlan_rcvmsg` [static]

Definition at line 50 of file `ng_vlan.c`.

7.164.3.9 `ng_shutdown_t ng_vlan_shutdown` [static]

Definition at line 51 of file `ng_vlan.c`.

7.164.3.10 `struct ng_parse_array_info ng_vlan_table_array_info` [static]**Initial value:**

```
{
    &ng_vlan_filter_type,
    ng_vlan_getTableLength
}
```

Definition at line 75 of file `ng_vlan.c`.

7.164.3.11 `struct ng_parse_type ng_vlan_table_array_type` [static]**Initial value:**

```
{
    &ng_parse_array_type,
    &ng_vlan_table_array_info
}
```

Definition at line 79 of file `ng_vlan.c`.

7.164.3.12 `struct ng_parse_struct_field ng_vlan_table_fields[]` [static]**Initial value:**

```
NG_VLAN_TABLE_FIELDS
```

Definition at line 83 of file `ng_vlan.c`.

7.164.3.13 `struct ng_parse_type ng_vlan_table_type` [static]**Initial value:**

```
{
    &ng_parse_struct_type,
    &ng_vlan_table_fields
}
```

Definition at line 85 of file `ng_vlan.c`.

7.164.3.14 struct ng_type ng_vlan_t [static]**Initial value:**

```
{
    .version =      NG_ABI_VERSION,
    .name =        NG_VLAN_NODE_TYPE,
    .constructor = ng_vlan_constructor,
    .rcvmsg =      ng_vlan_rcvmsg,
    .shutdown =    ng_vlan_shutdown,
    .newhook =     ng_vlan_newhook,
    .rcvdata =     ng_vlan_rcvdata,
    .disconnect =  ng_vlan_disconnect,
    .cmdlist =     ng_vlan_cmdlist,
}
```

Definition at line 116 of file ng_vlan.c.

7.165 /usr/src/sys/netgraph/ng_vlan.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ng_vlan_filter](#)
- struct [ng_vlan_table](#)

Defines

- #define [NG_VLAN_NODE_TYPE](#) "vlan"
- #define [NGM_VLAN_COOKIE](#) 1068486472
- #define [NG_VLAN_HOOK_DOWNSTREAM](#) "downstream"
- #define [NG_VLAN_HOOK_NOMATCH](#) "nomatch"
- #define [NG_VLAN_FILTER_FIELDS](#)
- #define [NG_VLAN_TABLE_FIELDS](#)

Enumerations

- enum { [NGM_VLAN_ADD_FILTER](#) = 1, [NGM_VLAN_DEL_FILTER](#), [NGM_VLAN_GET_TABLE](#) }

7.165.1 Define Documentation

7.165.1.1 #define NG_VLAN_FILTER_FIELDS

Value:

```

{
    { "hook",          &ng_parse_hookbuf_type },          \
    { "vlan",         &ng_parse_uint16_type },          \
    { NULL }
}

```

Definition at line 56 of file `ng_vlan.h`.

7.165.1.2 #define NG_VLAN_HOOK_DOWNSTREAM "downstream"

Definition at line 39 of file `ng_vlan.h`.

Referenced by `ng_vlan_newhook()`.

7.165.1.3 #define NG_VLAN_HOOK_NOMATCH "nomatch"

Definition at line 40 of file `ng_vlan.h`.

Referenced by `ng_vlan_newhook()`.

7.165.1.4 #define NG_VLAN_NODE_TYPE "vlan"

Definition at line 35 of file ng_vlan.h.

7.165.1.5 #define NG_VLAN_TABLE_FIELDS

Value:

```
{
    { "n",          &ng_parse_uint32_type },          \
    { "filter",    &ng_vlan_table_array_type },      \
    { NULL }
}
```

Definition at line 69 of file ng_vlan.h.

7.165.1.6 #define NGM_VLAN_COOKIE 1068486472

Definition at line 36 of file ng_vlan.h.

Referenced by ng_vlan_rcvmsg().

7.165.2 Enumeration Type Documentation

7.165.2.1 anonymous enum

Enumerator:

```
NGM_VLAN_ADD_FILTER
NGM_VLAN_DEL_FILTER
NGM_VLAN_GET_TABLE
```

Definition at line 43 of file ng_vlan.h.

Index

- /usr/* Directory Reference, 40
- /usr/src/* Directory Reference, 33
- /usr/src/sys/* Directory Reference, 36
- /usr/src/sys/netgraph/* Directory Reference, 29
- /usr/src/sys/netgraph/atm/* Directory Reference, 17
- /usr/src/sys/netgraph/atm/atmpif/* Directory Reference, 18
- /usr/src/sys/netgraph/atm/atmpif/ng_atmpif.c*, 532
- /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_harp.c*, 540
- /usr/src/sys/netgraph/atm/atmpif/ng_atmpif_var.h*, 546
- /usr/src/sys/netgraph/atm/ccatm/* Directory Reference, 21
- /usr/src/sys/netgraph/atm/ccatm/ng_ccatm.c*, 552
- /usr/src/sys/netgraph/atm/ccatm/ng_ccatm_cust.h*, 565
- /usr/src/sys/netgraph/atm/ng_atm.c*, 567
- /usr/src/sys/netgraph/atm/ng_atm.h*, 583
- /usr/src/sys/netgraph/atm/ng_atmpif.h*, 588
- /usr/src/sys/netgraph/atm/ng_ccatm.h*, 593
- /usr/src/sys/netgraph/atm/ng_sscfu.h*, 597
- /usr/src/sys/netgraph/atm/ng_sscop.h*, 599
- /usr/src/sys/netgraph/atm/ng_uni.h*, 602
- /usr/src/sys/netgraph/atm/ngatmbase.c*, 605
- /usr/src/sys/netgraph/atm/ngatmbase.h*, 609
- /usr/src/sys/netgraph/atm/sscfu/* Directory Reference, 34
- /usr/src/sys/netgraph/atm/sscfu/ng_sscfu.c*, 611
- /usr/src/sys/netgraph/atm/sscfu/ng_sscfu_cust.h*, 618
- /usr/src/sys/netgraph/atm/sscop/* Directory Reference, 35
- /usr/src/sys/netgraph/atm/sscop/ng_sscop.c*, 621
- /usr/src/sys/netgraph/atm/sscop/ng_sscop_cust.h*, 630
- /usr/src/sys/netgraph/atm/uni/* Directory Reference, 39
- /usr/src/sys/netgraph/atm/uni/ng_uni.c*, 639
- /usr/src/sys/netgraph/atm/uni/ng_uni_cust.h*, 651
- /usr/src/sys/netgraph/bluetooth/* Directory Reference, 19
- /usr/src/sys/netgraph/bluetooth/common/* Directory Reference, 22
- /usr/src/sys/netgraph/bluetooth/common/ng_bluetooth.c*, 657
- /usr/src/sys/netgraph/bluetooth/drivers/* Directory Reference, 23
- /usr/src/sys/netgraph/bluetooth/drivers/bt3c/* Directory Reference, 20
- /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_pccard.c*, 662
- /usr/src/sys/netgraph/bluetooth/drivers/bt3c/ng_bt3c_var.h*, 674
- /usr/src/sys/netgraph/bluetooth/drivers/h4/* Directory Reference, 24
- /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4.c*, 678
- /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_prse.h*, 687
- /usr/src/sys/netgraph/bluetooth/drivers/h4/ng_h4_var.h*, 688
- /usr/src/sys/netgraph/bluetooth/drivers/ubt/* Directory Reference, 37
- /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt.c*, 691
- /usr/src/sys/netgraph/bluetooth/drivers/ubt/ng_ubt_var.h*, 705
- /usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/* Directory Reference, 38
- /usr/src/sys/netgraph/bluetooth/drivers/ubtbcmfw/ubtbcmfw.c*, 710
- /usr/src/sys/netgraph/bluetooth/hci/* Directory Reference, 25
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.c*, 718
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_cmds.h*, 726
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.c*, 729
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_evnt.h*, 739
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_main.c*, 741
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.c*, 750
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_misc.h*, 756
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_prse.h*,

- 761
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.c, 764
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_ulpi.h, 771
- /usr/src/sys/netgraph/bluetooth/hci/ng_hci_var.h, 776
- /usr/src/sys/netgraph/bluetooth/include/ Directory Reference, 26
- /usr/src/sys/netgraph/bluetooth/include/ng_bluetooth.h, 783
- /usr/src/sys/netgraph/bluetooth/include/ng_bt3c.h, 791
- /usr/src/sys/netgraph/bluetooth/include/ng_btsocket.h, 795
- /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_hci_raw.h, 805
- /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_l2cap.h, 811
- /usr/src/sys/netgraph/bluetooth/include/ng_btsocket_rfcomm.h, 823
- /usr/src/sys/netgraph/bluetooth/include/ng_h4.h, 843
- /usr/src/sys/netgraph/bluetooth/include/ng_hci.h, 847
- /usr/src/sys/netgraph/bluetooth/include/ng_l2cap.h, 900
- /usr/src/sys/netgraph/bluetooth/include/ng_ubt.h, 919
- /usr/src/sys/netgraph/bluetooth/l2cap/ Directory Reference, 27
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.c, 922
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_cmds.h, 927
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evt.c, 934
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_evt.h, 944
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.c, 946
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_llpi.h, 953
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_main.c, 959
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.c, 968
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_misc.h, 977
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_prse.h, 985
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.c, 986
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_ulpi.h, 997
- /usr/src/sys/netgraph/bluetooth/l2cap/ng_l2cap_var.h, 1006
- /usr/src/sys/netgraph/bluetooth/socket/ Directory Reference, 32
- /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket.c, 1010
- /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_hci_raw.c, 1016
- /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap.c, 1029
- /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_l2cap_raw.c, 1051
- /usr/src/sys/netgraph/bluetooth/socket/ng_btsocket_rfcomm.c, 1063
- /usr/src/sys/netgraph/netflow/ Directory Reference, 28
- /usr/src/sys/netgraph/netflow/netflow.c, 1090
- /usr/src/sys/netgraph/netflow/netflow.h, 1098
- /usr/src/sys/netgraph/netflow/ng_netflow.c, 1100
- /usr/src/sys/netgraph/netflow/ng_netflow.h, 1108
- /usr/src/sys/netgraph/netgraph.h, 1116
- /usr/src/sys/netgraph/ng_UI.c, 1656
- /usr/src/sys/netgraph/ng_UI.h, 1660
- /usr/src/sys/netgraph/ng_async.c, 1158
- /usr/src/sys/netgraph/ng_async.h, 1165
- /usr/src/sys/netgraph/ng_atmllc.c, 1168
- /usr/src/sys/netgraph/ng_atmllc.h, 1173
- /usr/src/sys/netgraph/ng_base.c, 1175
- /usr/src/sys/netgraph/ng_bpf.c, 1204
- /usr/src/sys/netgraph/ng_bpf.h, 1211
- /usr/src/sys/netgraph/ng_bridge.c, 1213
- /usr/src/sys/netgraph/ng_bridge.h, 1225
- /usr/src/sys/netgraph/ng_cisco.c, 1228
- /usr/src/sys/netgraph/ng_cisco.h, 1237
- /usr/src/sys/netgraph/ng_deflate.c, 1240
- /usr/src/sys/netgraph/ng_deflate.h, 1247
- /usr/src/sys/netgraph/ng_device.c, 1249
- /usr/src/sys/netgraph/ng_device.h, 1256
- /usr/src/sys/netgraph/ng_echo.c, 1257
- /usr/src/sys/netgraph/ng_echo.h, 1260
- /usr/src/sys/netgraph/ng_eiface.c, 1261
- /usr/src/sys/netgraph/ng_eiface.h, 1268
- /usr/src/sys/netgraph/ng_etf.c, 1270
- /usr/src/sys/netgraph/ng_etf.h, 1276
- /usr/src/sys/netgraph/ng_ether.c, 1278
- /usr/src/sys/netgraph/ng_ether.h, 1286
- /usr/src/sys/netgraph/ng_fec.c, 1288
- /usr/src/sys/netgraph/ng_fec.h, 1297
- /usr/src/sys/netgraph/ng_frame_relay.c, 1300
- /usr/src/sys/netgraph/ng_frame_relay.h, 1307
- /usr/src/sys/netgraph/ng_gif.c, 1308
- /usr/src/sys/netgraph/ng_gif.h, 1316

- [/usr/src/sys/netgraph/ng_gif_demux.c](#), 1318
- [/usr/src/sys/netgraph/ng_gif_demux.h](#), 1324
- [/usr/src/sys/netgraph/ng_hole.c](#), 1326
- [/usr/src/sys/netgraph/ng_hole.h](#), 1330
- [/usr/src/sys/netgraph/ng_hub.c](#), 1332
- [/usr/src/sys/netgraph/ng_hub.h](#), 1335
- [/usr/src/sys/netgraph/ng_iface.c](#), 1336
- [/usr/src/sys/netgraph/ng_iface.h](#), 1345
- [/usr/src/sys/netgraph/ng_ip_input.c](#), 1347
- [/usr/src/sys/netgraph/ng_ip_input.h](#), 1350
- [/usr/src/sys/netgraph/ng_ipfw.c](#), 1351
- [/usr/src/sys/netgraph/ng_ipfw.h](#), 1357
- [/usr/src/sys/netgraph/ng_ksocket.c](#), 1359
- [/usr/src/sys/netgraph/ng_ksocket.h](#), 1372
- [/usr/src/sys/netgraph/ng_l2tp.c](#), 1374
- [/usr/src/sys/netgraph/ng_l2tp.h](#), 1389
- [/usr/src/sys/netgraph/ng_lmi.c](#), 1393
- [/usr/src/sys/netgraph/ng_lmi.h](#), 1403
- [/usr/src/sys/netgraph/ng_message.h](#), 1406
- [/usr/src/sys/netgraph/ng_mppc.c](#), 1419
- [/usr/src/sys/netgraph/ng_mppc.h](#), 1427
- [/usr/src/sys/netgraph/ng_nat.c](#), 1430
- [/usr/src/sys/netgraph/ng_nat.h](#), 1435
- [/usr/src/sys/netgraph/ng_one2many.c](#), 1437
- [/usr/src/sys/netgraph/ng_one2many.h](#), 1443
- [/usr/src/sys/netgraph/ng_parse.c](#), 1446
- [/usr/src/sys/netgraph/ng_parse.h](#), 1475
- [/usr/src/sys/netgraph/ng_ppp.c](#), 1483
- [/usr/src/sys/netgraph/ng_ppp.h](#), 1511
- [/usr/src/sys/netgraph/ng_pppoe.c](#), 1517
- [/usr/src/sys/netgraph/ng_pppoe.h](#), 1530
- [/usr/src/sys/netgraph/ng_pptpgre.c](#), 1539
- [/usr/src/sys/netgraph/ng_pptpgre.h](#), 1551
- [/usr/src/sys/netgraph/ng_pred1.c](#), 1554
- [/usr/src/sys/netgraph/ng_pred1.h](#), 1562
- [/usr/src/sys/netgraph/ng_rfc1490.c](#), 1564
- [/usr/src/sys/netgraph/ng_rfc1490.h](#), 1571
- [/usr/src/sys/netgraph/ng_sample.c](#), 1573
- [/usr/src/sys/netgraph/ng_sample.h](#), 1578
- [/usr/src/sys/netgraph/ng_socket.c](#), 1580
- [/usr/src/sys/netgraph/ng_socket.h](#), 1593
- [/usr/src/sys/netgraph/ng_socketvar.h](#), 1595
- [/usr/src/sys/netgraph/ng_source.c](#), 1596
- [/usr/src/sys/netgraph/ng_source.h](#), 1605
- [/usr/src/sys/netgraph/ng_split.c](#), 1607
- [/usr/src/sys/netgraph/ng_split.h](#), 1611
- [/usr/src/sys/netgraph/ng_sppp.c](#), 1612
- [/usr/src/sys/netgraph/ng_sppp.h](#), 1619
- [/usr/src/sys/netgraph/ng_tag.c](#), 1621
- [/usr/src/sys/netgraph/ng_tag.h](#), 1628
- [/usr/src/sys/netgraph/ng_tcpmss.c](#), 1630
- [/usr/src/sys/netgraph/ng_tcpmss.h](#), 1636
- [/usr/src/sys/netgraph/ng_tee.c](#), 1638
- [/usr/src/sys/netgraph/ng_tee.h](#), 1643
- [/usr/src/sys/netgraph/ng_tty.c](#), 1645
- [/usr/src/sys/netgraph/ng_tty.h](#), 1654
- [/usr/src/sys/netgraph/ng_vjc.c](#), 1661
- [/usr/src/sys/netgraph/ng_vjc.h](#), 1668
- [/usr/src/sys/netgraph/ng_vlan.c](#), 1671
- [/usr/src/sys/netgraph/ng_vlan.h](#), 1678
- [_NGI_ARG1](#)
netgraph.h, 1125
- [_NGI_ARG2](#)
netgraph.h, 1125
- [_NGI_CLR_HOOK](#)
netgraph.h, 1125
- [_NGI_CLR_NODE](#)
netgraph.h, 1125
- [_NGI_FN](#)
netgraph.h, 1126
- [_NGI_HOOK](#)
netgraph.h, 1126
- [_NGI_M](#)
netgraph.h, 1126
- [_NGI_META](#)
netgraph.h, 1126
- [_NGI_MSG](#)
netgraph.h, 1126
- [_NGI_NODE](#)
netgraph.h, 1126
- [_NGI_RETADDR](#)
netgraph.h, 1126
- [_NGI_SET_HOOK](#)
netgraph.h, 1126
- [_NGI_SET_NODE](#)
netgraph.h, 1127
- [_NG_ABI_VERSION](#)
netgraph.h, 1122
- [_NG_ALLOC_HOOK](#)
ng_base.c, 1179
- [_NG_ALLOC_NODE](#)
ng_base.c, 1179
- [_NG_HOOK_FORCE_QUEUE](#)
netgraph.h, 1122
- [_NG_HOOK_FORCE_WRITER](#)
netgraph.h, 1122
- [_NG_HOOK_IS_VALID](#)
netgraph.h, 1122
- [_NG_HOOK_NAME](#)
netgraph.h, 1122
- [_NG_HOOK_NODE](#)
netgraph.h, 1123
- [_NG_HOOK_NOT_VALID](#)
netgraph.h, 1123
- [_NG_HOOK_PEER](#)
netgraph.h, 1123
- [_NG_HOOK_PRIVATE](#)
netgraph.h, 1123

- _NG_HOOK_REF
 - netgraph.h, 1123
- _NG_HOOK_SET_PRIVATE
 - netgraph.h, 1123
- _NG_HOOK_SET_RCVDATA
 - netgraph.h, 1123
- _NG_HOOK_SET_RCVMSG
 - netgraph.h, 1123
- _NG_HOOK_UNREF
 - netgraph.h, 1123
- _NG_NODE_FORCE_WRITER
 - netgraph.h, 1123
- _NG_NODE_FOREACH_HOOK
 - netgraph.h, 1124
- _NG_NODE_HAS_NAME
 - netgraph.h, 1124
- _NG_NODE_ID
 - netgraph.h, 1124
- _NG_NODE_IS_VALID
 - netgraph.h, 1124
- _NG_NODE_NAME
 - netgraph.h, 1124
- _NG_NODE_NOT_VALID
 - netgraph.h, 1124
- _NG_NODE_NUMHOOKS
 - netgraph.h, 1124
- _NG_NODE_PRIVATE
 - netgraph.h, 1124
- _NG_NODE_REALLY_DIE
 - netgraph.h, 1124
- _NG_NODE_REF
 - netgraph.h, 1125
- _NG_NODE_REVIVE
 - netgraph.h, 1125
- _NG_NODE_SET_PRIVATE
 - netgraph.h, 1125
- _NG_NODE_UNREF
 - netgraph.h, 1125
- _TIMER_DESTROY
 - ng_uni_cust.h, 652
- _TIMER_INIT
 - ng_uni_cust.h, 652
- _TIMER_START
 - ng_uni_cust.h, 652
- _TIMER_STOP
 - ng_uni_cust.h, 653
- __FBSDID
 - ng_atm.c, 571
 - ng_atmpif.c, 535
 - ng_atmpif_harp.c, 542
 - ng_ccatm.c, 555
 - ng_source.c, 1599
 - ng_sppp.c, 1615
 - ng_sscfu.c, 613
 - ng_sscop.c, 624
 - ng_uni.c, 642
 - ngatmbase.c, 606
- __attribute__
 - 41
 - accept_role_switch, 43
 - attempt, 43
 - auth_enable, 43
 - b, 43
 - bdaddr, 43
 - clock_offset, 44
 - code, 44
 - command, 44
 - con_handle, 44
 - condition, 44
 - counter, 44
 - country_code, 44
 - dcid, 44
 - delay_variation, 44
 - delete_all, 44
 - encryption_enable, 45
 - encryption_mode, 45
 - event, 45
 - event_mask, 45
 - features, 45
 - filter_condition_type, 45
 - filter_type, 45
 - flags, 45
 - flow_control, 45
 - h2hc_flow, 45
 - hardware_code, 46
 - hci_revision, 46
 - hci_version, 46
 - hold_mode_activity, 46
 - ident, 46
 - inquiry_length, 46
 - inquiry_scan_interval, 46
 - inquiry_scan_window, 46
 - interval, 46
 - key, 46
 - key_flag, 46
 - key_type, 47
 - lap, 47
 - latency, 47
 - lbmode, 47
 - lcid, 47
 - length, 47
 - level, 47
 - link_type, 47
 - lmp_max_slots, 47
 - lmp_subversion, 47
 - lmp_version, 48
 - manufacturer, 48
 - max_acl_size, 48
 - max_interval, 48

- max_num_keys, 48
- max_period_length, 48
- max_sco_size, 48
- min_interval, 48
- min_period_length, 48
- name, 48
- num_acl_pkt, 48
- num_cmd_pkts, 49
- num_con_handles, 49
- num_iac, 49
- num_keys, 49
- num_keys_deleted, 49
- num_keys_read, 49
- num_keys_write, 49
- num_keys_written, 49
- num_responses, 49
- num_sco_pkt, 49
- opcode, 49
- page_scan_interval, 50
- page_scan_mode, 50
- page_scan_period_mode, 50
- page_scan_rep_mode, 50
- page_scan_window, 50
- peak_bandwidth, 50
- pin, 50
- pin_size, 50
- pin_type, 50
- pkt_type, 50
- psm, 50
- quality, 51
- read_all, 51
- reason, 51
- result, 51
- role, 51
- rsi, 51
- scan_enable, 51
- scid, 51
- service_type, 51
- settings, 51
- status, 52
- timeout, 52
- token, 52
- token_bucket_size, 52
- token_rate, 52
- type, 52
- uclass, 52
- unit_mode, 52
- __packed
 - ng_pppoe.h, 1538
- __packed__
 - netflow.h, 1099
- __pad
 - vatmpif_header, 522
- _ng_l2cap_build_cfg_options
 - ng_l2cap_cmds.h, 927
- _ng_l2cap_cfg_req
 - ng_l2cap_cmds.h, 927
- _ng_l2cap_cfg_rsp
 - ng_l2cap_cmds.h, 927
- _ng_l2cap_cmd_rej
 - ng_l2cap_cmds.h, 928
- _ng_l2cap_con_req
 - ng_l2cap_cmds.h, 928
- _ng_l2cap_con_rsp
 - ng_l2cap_cmds.h, 928
- _ng_l2cap_discon_req
 - ng_l2cap_cmds.h, 929
- _ng_l2cap_discon_rsp
 - ng_l2cap_cmds.h, 929
- _ng_l2cap_echo_req
 - ng_l2cap_cmds.h, 930
- _ng_l2cap_info_req
 - ng_l2cap_cmds.h, 930
- _ng_l2cap_info_rsp
 - ng_l2cap_cmds.h, 931
- aa_dropped
 - stats, 505
- aa_signals
 - stats, 505
- aal
 - ngm_atm_cpcs_init, 426
 - vatmpif_header, 522
- aal5_crc_len
 - hva_stats_aal5, 90
- aal5_drops
 - hva_stats_aal5, 90
- aal5_pdu_crc
 - hva_stats_aal5, 90
- aal5_pdu_drops
 - hva_stats_aal5, 90
- aal5_pdu_errs
 - hva_stats_aal5, 90
- aal5_pdu_rcvd
 - hva_stats_aal5, 90
- aal5_pdu_xmit
 - hva_stats_aal5, 90
- aal5_rcvd
 - hva_stats_aal5, 91
- aal5_xmit
 - hva_stats_aal5, 91
- abuf
 - ng_async_private, 117
- ac_name
 - sess_neg, 494
- ac_name_len
 - sess_neg, 494
- accept_role_switch

- __attribute__, 43
- acem
 - ng_async_cfg, 116
- ack_timer
 - rftcomm_mcc_pn, 485
- ackp
 - ng_pptpgre_private, 371
- acks
 - l2tp_seq, 99
- acl
 - ng_hci_unit, 236
- acl_free
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- acl_pkts
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- acl_recv
 - ng_hci_node_stat_ep, 231
- acl_sent
 - ng_hci_node_stat_ep, 231
- acl_size
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- acr
 - ngm_atm_acr_change, 424
- ACTIVE_TIMEOUT
 - netflow.h, 1098
- active_timeout
 - ng_netflow_settimeouts, 337
- activeLinks
 - ng_ppp_private, 365
- activeMany
 - ng_one2many_private, 344
- ADD_BYTE
 - ng_async.c, 1159
- addr
 - ng_bridge_host, 132
 - ng_btssocket_hci_raw_pcb, 160
 - ng_ksocket_accept, 256
 - ngm_ccatm_addr_req, 434
 - ngm_ccatm_get_addresses, 435
- ADDR_HASH
 - netflow.c, 1092
- address
 - cisco_header, 63
 - rftcomm_cmd_hdr, 481
 - rftcomm_frame_hdr, 482
 - rftcomm_mcc_msc, 484
 - rftcomm_mcc_rls, 487
- addrlen
 - frmrel_softc, 83
- adtf
 - ngm_atm_cpcs_init, 426
- af
 - protoent, 480
- age
 - ng_bridge_host, 132
- AGED
 - netflow.c, 1092
- ALIGNMENT
 - ng_parse.c, 1451
- alignment
 - ng_parse_struct_field, 349
- all
 - flow_rec, 81
- allLinksEqual
 - ng_ppp_private, 365
- ALOT
 - ng_btssocket_rftcomm.c, 1066
- ALT
 - frmrel_softc, 83
- amode
 - ng_async_private, 118
- amru
 - ng_async_cfg, 116
- ANNEXA
 - ng_lmi.c, 1395
- ANNEXD
 - ng_lmi.c, 1395
- apply
 - ng_item, 254
- arg
 - sscop_arg, 502
- arglen
 - ng_mesg::ng_msghdr, 321
- ASSERT
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 632
 - ng_uni_cust.h, 653
- async
 - ng_async_private, 118
- ASYNC_BUF_SIZE
 - ng_async.c, 1159
- asyncBadCheckSums
 - ng_async_stat, 120
- asyncFrames
 - ng_async_stat, 120
- asyncOctets
 - ng_async_stat, 120
- asyncOverflows
 - ng_async_stat, 120
- asyncRunts
 - ng_async_stat, 120
- atalk
 - cisco_priv, 67
- atm
 - ng_atmllc_priv, 122

- atm_rcvd
 - hva_stats_atm, 92
- atm_xmit
 - hva_stats_atm, 92
- atmmmedia
 - ng_atm.c, 576
- ato
 - ng_pptpgre_ackp, 367
- attempt
 - __attribute__, 43
- auth_enable
 - __attribute__, 43
- autod
 - nglmistat, 423
- autoSrcAddr
 - private, 475
- aux
 - ng_l2cap_cmd, 270
- b
 - __attribute__, 43
- badProtos
 - ng_ppp_link_stat, 361
- bandwidth
 - ng_ppp_link_conf, 359
- bdaddr
 - __attribute__, 43
 - ng_btsocket_hci_raw_node_bdaddr, 149
 - ng_hci_lp_con_cfm_ep, 212
 - ng_hci_lp_con_ind_ep, 213
 - ng_hci_lp_con_req_ep, 214
 - ng_hci_lp_con_rsp_ep, 215
 - ng_hci_neighbor, 222
 - ng_hci_node_con_ep, 226
 - ng_hci_node_neighbor_cache_entry_ep, 230
 - ng_hci_node_up_ep, 233
 - ng_hci_unit, 236
 - ng_hci_unit_con, 242
 - ng_l2cap, 262
 - ng_l2cap_l2ca_con_ind_ip, 282
 - ng_l2cap_l2ca_con_ip, 283
 - ng_l2cap_l2ca_con_rsp_ip, 285
 - ng_l2cap_l2ca_get_info_ip, 290
 - ng_l2cap_l2ca_grp_add_member_ip, 292
 - ng_l2cap_l2ca_ping_ip, 299
 - ng_l2cap_l2ca_ping_op, 300
 - ng_l2cap_l2ca_qos_ind_ip, 301
- bdaddr_p
 - ng_hci.h, 896
- bit_decl
 - ng_btsocket_hci_raw_filter, 148
 - ng_btsocket_hci_raw_sec_filter, 162
- bit_rate
 - rfcomm_mcc_rpn, 488
- bits
 - ng_mppc_config, 324
- block_address
 - ng_bt3c_firmware_block_ep, 141
- block_alignment
 - ng_bt3c_firmware_block_ep, 141
- block_size
 - ng_bt3c_firmware_block_ep, 141
- bluetooth_hci_command_timeout
 - ng_bluetooth.c, 658
 - ng_bluetooth.h, 789
- bluetooth_hci_command_timeout_value
 - ng_bluetooth.c, 660
- bluetooth_hci_connect_timeout
 - ng_bluetooth.c, 658
 - ng_bluetooth.h, 789
- bluetooth_hci_connect_timeout_value
 - ng_bluetooth.c, 660
- bluetooth_hci_max_neighbor_age
 - ng_bluetooth.c, 658
 - ng_bluetooth.h, 789
- bluetooth_hci_max_neighbor_age_value
 - ng_bluetooth.c, 660
- bluetooth_l2cap_ertx_timeout
 - ng_bluetooth.c, 658
 - ng_bluetooth.h, 790
- bluetooth_l2cap_ertx_timeout_value
 - ng_bluetooth.c, 661
- bluetooth_l2cap_rtx_timeout
 - ng_bluetooth.c, 659
 - ng_bluetooth.h, 790
- bluetooth_l2cap_rtx_timeout_value
 - ng_bluetooth.c, 661
- bluetooth_mod
 - ng_bluetooth.c, 661
- bluetooth_modevent
 - ng_bluetooth.c, 659
- BLUETOOTH_PROTO_HCI
 - ng_btsocket.h, 797
- BLUETOOTH_PROTO_L2CAP
 - ng_btsocket.h, 797
- BLUETOOTH_PROTO_RFCOMM
 - ng_btsocket.h, 797
- bluetooth_set_hci_command_timeout_value
 - ng_bluetooth.c, 659
- bluetooth_set_hci_connect_timeout_value
 - ng_bluetooth.c, 659
- bluetooth_set_l2cap_ertx_timeout_value
 - ng_bluetooth.c, 659
- bluetooth_set_l2cap_rtx_timeout_value
 - ng_bluetooth.c, 659
- body
 - ng_item, 254
- both

- flow_rec, 81
- bpf_prog
 - ng_bpf_hookprog, 127
- bpf_prog_len
 - ng_bpf_hookprog, 127
- BT3C_ADDR_H
 - ng_bt3c_var.h, 674
- BT3C_ADDR_L
 - ng_bt3c_var.h, 674
- BT3C_ANTENNA_OUT
 - ng_bt3c_var.h, 675
- BT3C_CONTROL
 - ng_bt3c_var.h, 675
- BT3C_DATA_H
 - ng_bt3c_var.h, 675
- BT3C_DATA_L
 - ng_bt3c_var.h, 675
- BT3C_DEFAULTQLEN
 - ng_bt3c_var.h, 675
- bt3c_devclass
 - ng_bt3c_pccard.c, 671
- bt3c_download_firmware
 - ng_bt3c_pccard.c, 666
- BT3C_FIFO_SIZE
 - ng_bt3c_var.h, 675
- bt3c_forward
 - ng_bt3c_pccard.c, 666
- bt3c_intr
 - ng_bt3c_pccard.c, 667
- bt3c_modevent
 - ng_bt3c_pccard.c, 667
- bt3c_pccard_attach
 - ng_bt3c_pccard.c, 667
- bt3c_pccard_detach
 - ng_bt3c_pccard.c, 668
- bt3c_pccard_driver
 - ng_bt3c_pccard.c, 671
- bt3c_pccard_methods
 - ng_bt3c_pccard.c, 671
- bt3c_pccard_probe
 - ng_bt3c_pccard.c, 668
- bt3c_read
 - ng_bt3c_pccard.c, 665
- bt3c_read_control
 - ng_bt3c_pccard.c, 665
- bt3c_read_data
 - ng_bt3c_pccard.c, 665
- bt3c_receive
 - ng_bt3c_pccard.c, 668
- bt3c_send
 - ng_bt3c_pccard.c, 668
- bt3c_set_address
 - ng_bt3c_pccard.c, 665
- bt3c_softc, 54
 - debug, 55
 - dev, 55
 - flags, 55
 - hook, 55
 - inq, 55
 - iobase, 55
 - iobase_rid, 55
 - ioh, 55
 - iot, 56
 - irq, 56
 - irq_cookie, 56
 - irq_rid, 56
 - ith, 56
 - m, 56
 - node, 56
 - outq, 56
 - stat, 57
 - state, 57
 - status, 57
 - want, 57
- bt3c_softc_p
 - ng_bt3c_var.h, 677
- bt3c_softc_t
 - ng_bt3c_var.h, 677
- bt3c_swi_intr
 - ng_bt3c_pccard.c, 668
- bt3c_write
 - ng_bt3c_pccard.c, 665
- bt3c_write_control
 - ng_bt3c_pccard.c, 666
- bt3c_write_data
 - ng_bt3c_pccard.c, 666
- BT3C_XMIT
 - ng_bt3c_var.h, 675
- buffer
 - ng_btsocket_hci_raw_node_buffer, 150
 - ng_hci_unit, 236
- bufSize
 - ng_parse_fixedstring_info, 348
- bund
 - ng_ppp_node_conf, 364
- bundleStats
 - ng_ppp_private, 365
- BYTE1_C_R
 - ng_frame_relay.c, 1301
- BYTE2_BEEN
 - ng_frame_relay.c, 1301
- BYTE2_DE
 - ng_frame_relay.c, 1301
- BYTE2_FECN
 - ng_frame_relay.c, 1302
- bytes
 - flow_entry_data, 77
 - packet, 462

- uniq, [521](#)
- bytes_recv
 - ng_bt3c_node_stat_ep, [143](#)
 - ng_h4_node_stat_ep, [211](#)
 - ng_hci_node_stat_ep, [231](#)
 - ng_ubt_node_stat_ep, [407](#)
- bytes_sent
 - ng_bt3c_node_stat_ep, [143](#)
 - ng_h4_node_stat_ep, [211](#)
 - ng_hci_node_stat_ep, [231](#)
 - ng_ubt_node_stat_ep, [407](#)
- bytesInQueue
 - ng_ppp_link, [357](#)
- BYTEX_EA
 - ng_frame_relay.c, [1302](#)
- c
 - uni_timer, [519](#)
- CACHEHIGHWAT
 - ng_netflow.h, [1109](#)
- CACHELOWAT
 - ng_netflow.h, [1109](#)
- CACHESIZE
 - ng_netflow.h, [1109](#)
- CALL_ALLOC
 - ng_uni_cust.h, [653](#)
- CALL_FREE
 - ng_uni_cust.h, [653](#)
- carrier
 - ngm_atm_if_change, [430](#)
- cc
 - ng_mppc_dir, [325](#)
- CCASSERT
 - ng_ccatm_cust.h, [566](#)
- ccatm_op, [58](#)
 - data, [58](#)
 - op, [58](#)
- CCFREE
 - ng_ccatm_cust.h, [566](#)
- CCGETERRNO
 - ng_ccatm_cust.h, [566](#)
- cchhook, [59](#)
 - hook, [59](#)
 - inst, [59](#)
 - is_uni, [60](#)
 - node, [60](#)
- CCMALLOC
 - ng_ccatm_cust.h, [566](#)
- ccnode, [61](#)
 - data, [61](#)
 - dump, [61](#)
 - dump_first, [62](#)
 - dump_last, [62](#)
 - hook_cnt, [62](#)
 - manage, [62](#)
 - node, [62](#)
- CCZALLOC
 - ng_ccatm_cust.h, [566](#)
- cdf
 - ngm_atm_cpcs_init, [426](#)
- cellhdr
 - vatmpif_header, [522](#)
- cfc
 - ng_btsocket_rfcomm_fc_info, [180](#)
- cfg
 - ng_async_private, [118](#)
 - ng_deflate_private, [194](#)
 - ng_mppc_dir, [325](#)
 - ng_pred1_private, [376](#)
- cfg_state
 - ng_btsocket_l2cap_pcb, [164](#)
 - ng_l2cap_chan, [266](#)
- ch
 - ng_l2cap_cmd, [270](#)
- CHAN_ACTIVE
 - ng_frame_relay.c, [1302](#)
- CHAN_VALID
 - ng_frame_relay.c, [1302](#)
- chand
 - ngt_sc, [456](#)
- channel
 - frmrel_softc, [84](#)
 - ng_btsocket_rfcomm_pcb, [183](#)
 - XXX, [527](#)
 - XXX_hookinfo, [529](#)
- channels
 - ng_btsocket_l2cap_raw_chan_list, [169](#)
- CHECK_DATA_MBUF
 - ng_base.c, [1179](#)
- cid
 - greheader, [85](#)
 - ng_btsocket_l2cap_pcb, [164](#)
 - ng_l2cap, [262](#)
 - ng_l2cap_cmd_rej_data_t, [271](#)
 - ng_pptpgre_conf, [369](#)
- CISCO_ADDR_REPLY
 - ng_cisco.c, [1230](#)
- CISCO_ADDR_REQ
 - ng_cisco.c, [1230](#)
- cisco_constructor
 - ng_cisco.c, [1232](#), [1234](#)
- cisco_disconnect
 - ng_cisco.c, [1232](#), [1234](#)
- cisco_header, [63](#)
 - address, [63](#)
 - control, [63](#)
 - protocol, [63](#)
- CISCO_HEADER_LEN

- ng_cisco.c, 1230
- cisco_input
 - ng_cisco.c, 1232
- CISCO_KEEPALIVE
 - ng_cisco.c, 1231
- cisco_keepalive
 - ng_cisco.c, 1232
- CISCO_KEEPALIVE_REQ
 - ng_cisco.c, 1231
- CISCO_MULTICAST
 - ng_cisco.c, 1231
- cisco_newhook
 - ng_cisco.c, 1233, 1234
- cisco_notify
 - ng_cisco.c, 1233
- cisco_packet, 64
 - par1, 64
 - par2, 64
 - rel, 64
 - time0, 64
 - time1, 64
 - type, 64
- CISCO_PACKET_LEN
 - ng_cisco.c, 1231
- cisco_priv, 66
 - atalk, 67
 - downstream, 67
 - handle, 67
 - inet, 67
 - inet6, 67
 - ipx, 67
 - local_seq, 67
 - localip, 67
 - localmask, 67
 - node, 67
 - remote_seq, 67
 - seqRetries, 67
- cisco_rcvdata
 - ng_cisco.c, 1233, 1234
- cisco_rcvmsg
 - ng_cisco.c, 1233, 1234
- cisco_send
 - ng_cisco.c, 1234
- cisco_shutdown
 - ng_cisco.c, 1234
- CISCO_UNICAST
 - ng_cisco.c, 1231
- clock_offset
 - __attribute__, 44
 - ng_hci_neighbor, 222
 - ng_hci_node_neighbor_cache_entry_ep, 230
- close
 - ng_type, 404
- cmd
 - ng_cmdlist, 191
 - ng_mesg::ng_msghdr, 321
 - ng_pppoe.h, 1537
- cmd_free
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- cmd_sent
 - ng_hci_node_stat_ep, 231
- cmd_timo
 - ng_hci_unit, 236
- cmdlist
 - ng_type, 404
- cmdq
 - ng_hci_unit, 236
- cmdstr
 - ng_mesg::ng_msghdr, 322
- cnt
 - sscop_merr, 504
- code
 - __attribute__, 44
 - ng_l2cap_cmd, 270
 - pppoe_hdr, 466
- command
 - __attribute__, 44
- COMPAT_3COM
 - ng_pppoe.c, 1519
- COMPAT_DLINK
 - ng_pppoe.c, 1519
- complete_command
 - ng_hci_cmds.c, 720
- completed
 - ng_hci_sync_con_queue_ep, 234
- compress
 - ng_deflate_private, 194
 - ng_pred1_private, 376
- compressCID
 - ngm_vjc_config, 447
- comptype
 - ng_parse.c, 1452
- con
 - ng_l2cap_chan, 266
 - ng_l2cap_cmd, 270
- con_compl
 - ng_hci_evnt.c, 731
- con_handle
 - __attribute__, 44
 - ng_hci_lp_con_cfm_ep, 212
 - ng_hci_lp_discon_ind_ep, 216
 - ng_hci_lp_discon_req_ep, 217
 - ng_hci_lp_qos_cfm_ep, 218
 - ng_hci_lp_qos_ind_ep, 219
 - ng_hci_lp_qos_req_ep, 220
 - ng_hci_node_con_ep, 226
 - ng_hci_sync_con_queue_ep, 234

- ng_hci_unit_con, 242
- ng_l2cap_con, 273
- ng_l2cap_node_con_ep, 306
- con_req
 - ng_hci_evt.c, 731
- con_timo
 - ng_hci_unit_con, 242
 - ng_l2cap_con, 273
- condition
 - __attribute__, 44
- conf
 - ng_bridge_private, 139
 - ng_l2tp_hook_private, 310
 - ng_l2tp_private, 311
 - ng_one2many_private, 344
 - ng_ppp_link, 357
 - ng_ppp_private, 365
 - ng_pptpgre_private, 371
 - ng_vjc_private, 412
 - vatmpif_unit, 524
- config
 - ngm_uni_set_config, 446
- connect
 - ng_type, 404
- connections
 - ng_btsocket_hci_raw_con_list, 147
 - ng_btsocket_l2cap_raw_con_list, 170
- conq
 - ng_hci_unit_con, 242
- constructor
 - ng_type, 404
- context
 - ng_item, 254
- control
 - cisco_header, 63
 - rfcomm_cmd_hdr, 481
 - rfcomm_frame_hdr, 482
- control_dseq
 - ng_l2tp_sess_config, 314
- cookie
 - ng_cmdlist, 191
 - uni_arg, 518
 - vatmpif_header, 522
- correct_mss
 - ng_tcpmss.c, 1632
- count
 - netflow_v1_header, 106
 - netflow_v5_header, 111
 - ngm_ccatm_get_addresses, 435
- counter
 - __attribute__, 44
- country_code
 - __attribute__, 44
- cpcs_term
 - ng_atm.c, 571
- Crc16
 - ng_pred1.c, 1556
- Crc16Table
 - ng_pred1.c, 1559
- creator
 - sess_con, 492
- credits
 - rfcomm_mcc_pn, 485
- CT_ARRAY
 - ng_parse.c, 1452
- CT_FIXEDARRAY
 - ng_parse.c, 1452
- CT_STRUCT
 - ng_parse.c, 1452
- ctl
 - ng_l2cap, 262
- ctlsock
 - ngsock, 454
- ctrl
 - ng_l2tp_private, 311
- ctrlnode
 - ng_deflate_private, 194
 - ng_mppc_private, 327
 - ng_pred1_private, 376
- CTX_VALID
 - ng_frame_relay.c, 1302
- CTX_VALUE
 - ng_frame_relay.c, 1302
- ctxinfo, 69
 - dici, 69
 - flags, 69
 - hook, 69
- cur_pcr
 - ng_atmpif_link_status, 124
 - ng_vatmpif_hook, 410
- current
 - ngm_queue_state, 441
- cwnd
 - l2tp_seq, 99
- cx
 - ng_deflate_private, 194
- D
 - ng_uni.c, 642
- d_port
 - flow_rec, 81
 - netflow_v1_record, 107
 - netflow_v5_record, 113
- da_m
 - ng_item, 254
- data
 - ccatm_op, 58
 - ccnode, 61

- datatag, 71
- greheader, 85
- ng_mesg, 319
- ng_ppp_frag, 355
- ngpppoe_init_data, 451
- sscfu_arg, 501
- sscop_arg, 502
- sscop_marg, 503
- uni_arg, 518
- data_buffer_overflow
 - ng_hci_evt.c, 732
- data_delivered
 - stats, 505
- data_len
 - ngpppoe_init_data, 451
- datahooks
 - firmrel_softc, 84
- datasock
 - ngsock, 454
- datatag, 71
 - data, 71
 - hdr, 71
- DBG
 - ng_device.c, 1251
- dcid
 - __attribute__, 44
 - ng_l2cap_chan, 266
 - ng_l2cap_cmd_rej_data_t, 271
 - ng_l2cap_node_chan_ep, 303
- DDD
 - ng_sscop.c, 623
- debug
 - bt3c_softc, 55
 - ng_btsocket_hci_raw_node_debug, 151
 - ng_btsocket_l2cap_raw_node_debug, 172
 - ng_h4_info, 209
 - ng_hci_unit, 236
 - ng_l2cap, 262
 - ng_vatmpif_config, 409
- debug_hook
 - PPPoE, 463
- debughook
 - XXX, 527
- debugLevel
 - ng_bridge_config, 130
- DECL_MBUF_ALLOC
 - ng_sscop_cust.h, 632
- DECL_MSGQ_GET
 - ng_sscop_cust.h, 632
- DECL_SIGQ_GET
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 633
- DECLARE_MODULE
 - ng_base.c, 1183
 - ng_bluetooth.c, 659
 - ng_btsocket.c, 1012
 - ngatmbase.c, 606
- DEFAULT_LOOP_TIMEOUT
 - ng_bridge.c, 1216
- DEFAULT_MAX_STALENESS
 - ng_bridge.c, 1216
- DEFAULT_MIN_STABLE_AGE
 - ng_bridge.c, 1216
- DEFINE_PARSE_STRUCT_TYPE
 - ng_base.c, 1179, 1183
- DEFLATE_BUF_SIZE
 - ng_deflate.c, 1241
- DEFLATE_HDRLIN
 - ng_deflate.c, 1241
- delay_variation
 - __attribute__, 44
 - ng_hci_lp_qos_req_ep, 220
- delete_all
 - __attribute__, 44
- dev
 - bt3c_softc, 55
 - ng_pptpgre_ackp, 367
- dir
 - flow_rec, 81
 - ng_ipfw_tag, 252
- discardability
 - ng_tag_prio, 398
- discon_compl
 - ng_hci_evt.c, 732
- discon_timo
 - ng_l2cap, 262
- disconnect
 - ng_type, 404
- dlci
 - ctxinfo, 69
 - ng_btsocket_rfcomm_pcb, 183
 - rfcomm_mcc_pn, 485
 - rfcomm_mcc_rpn, 488
 - XXX_hookinfo, 529
- DLCI_DOWN
 - ng_lmi.c, 1395
- DLCI_NULL
 - ng_lmi.c, 1395
- dlci_state
 - nglmi_softc, 420
- DLCI_UP
 - ng_lmi.c, 1395
- dlt
 - ng_netflow_setdlt, 335
- downlink
 - ng_rfc1490_private, 382
 - ng_UI_private, 408
- downstream

- cisco_priv, 67
- frmrel_softc, 84
- downstream_hook
 - ETF, 72
 - priv_p, 473
 - XXX, 528
- DRIVER_MODULE
 - ng_bt3c_pccard.c, 669
 - ng_ubt.c, 694
 - ubtbcmfw.c, 714
- dropFragments
 - ng_ppp_link_stat, 361
- drops
 - ng_bt_itemq, 144
 - ng_bt_mbufq, 146
- drv
 - ng_hci_unit, 236
- dst
 - ng_btsocket_l2cap_pcb, 164
 - ng_btsocket_l2cap_raw_pcb, 174
 - ng_btsocket_rfcomm_pcb, 183
- dst_addr
 - netflow_v1_record, 107
 - netflow_v5_record, 113
- dst_as
 - netflow_v5_record, 113
- dst_mask
 - flow_entry_data, 77
 - netflow_v5_record, 113
- dummy_disconnect
 - ng_socket.c, 1583
- dump
 - ccnode, 61
- dump_first
 - ccnode, 62
- dump_last
 - ccnode, 62
- dump_saal_signal
 - ng_uni.c, 642
- dump_uni_msg
 - ng_uni.c, 642
- dupFragments
 - ng_ppp_link_stat, 361
- echo_data
 - ng_btsocket_l2cap_raw_ping, 177
- echo_size
 - ng_btsocket_l2cap_raw_ping, 177
 - ng_l2cap_l2ca_ping_ip, 299
 - ng_l2cap_l2ca_ping_op, 300
- eh
 - PPPoE, 463
 - pppoe_full_hdr, 465
- el_dest
 - ng_item, 254
- el_flags
 - ng_item, 254
- el_hook
 - ng_item, 254
- el_next
 - ng_item, 254
- elapsedTime
 - ng_source_stats, 384
- elementType
 - ng_parse_array_info, 346
 - ng_parse_fixedarray_info, 347
- enable
 - ng_deflate_config, 193
 - ng_l2cap_l2ca_enable_clt_ip, 289
 - ng_mppc_config, 324
 - ng_pred1_config, 375
- enable_dseq
 - ng_l2tp_sess_config, 314
- enableACFComp
 - ng_ppp_link_conf, 359
- enableAlwaysAck
 - ng_pptpgre_conf, 369
- enableAtalk
 - ng_ppp_bund_conf, 352
- enableComp
 - ngm_vjc_config, 447
- enableCompression
 - ng_ppp_bund_conf, 352
- enabled
 - ng_async_cfg, 116
 - ng_l2tp_config, 308
 - ng_pptpgre_conf, 369
 - priv, 470
- enableDecomp
 - ngm_vjc_config, 447
- enableDecompression
 - ng_ppp_bund_conf, 352
- enableDecryption
 - ng_ppp_bund_conf, 352
- enableDelayedAck
 - ng_pptpgre_conf, 369
- enabledLinks
 - ng_one2many_config, 341
- enableEncryption
 - ng_ppp_bund_conf, 352
- enableIP
 - ng_ppp_bund_conf, 352
- enableIPv6
 - ng_ppp_bund_conf, 353
- enableIPX
 - ng_ppp_bund_conf, 353
- enableLink
 - ng_ppp_link_conf, 359

- enableMultilink
 - ng_ppp_bund_conf, 353
- enableProtoComp
 - ng_ppp_link_conf, 359
- enableRoundRobin
 - ng_ppp_bund_conf, 353
- enableVJCompression
 - ng_ppp_bund_conf, 353
- enableVJDecompression
 - ng_ppp_bund_conf, 353
- enableWindowing
 - ng_pptpgre_conf, 369
- enc
 - ng_rfc1490_private, 382
- encryption_change
 - ng_hci_evt.c, 732
- encryption_enable
 - __attribute__, 45
- encryption_mode
 - __attribute__, 45
 - ng_hci_node_con_ep, 226
 - ng_hci_unit_con, 242
- endTime
 - ng_source_stats, 384
- engine_id
 - netflow_v5_header, 111
- engine_type
 - netflow_v5_header, 111
- entries
 - ng_btsocket_hci_raw_node_neighbor_cache, 155
 - ngnf_flows, 448
- err
 - sscop_merr, 504
- error
 - ng_sscop_setparam_resp, 392
 - ngsock, 455
- Errors
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- errors
 - stats, 505
- ERROUT
 - ng_async.c, 1159
 - ng_bpf.c, 1205
 - ng_deflate.c, 1242
 - ng_device.c, 1251
 - ng_ksocket.c, 1362
 - ng_mppc.c, 1421
 - ng_netflow.h, 1109
 - ng_ppp.c, 1487
 - ng_pptpgre.c, 1542
 - ng_pred1.c, 1555
 - ng_rfc1490.c, 1566
 - ng_tag.c, 1622
 - ng_tcpmss.c, 1631
 - ng_tty.c, 1647
 - ng_UI.c, 1657
 - ng_vjc.c, 1663
- ETF, 72
 - downstream_hook, 72
 - flags, 72
 - hashtable, 72
 - node, 73
 - nomatch_hook, 73
 - packets_in, 73
 - packets_out, 73
- ETF_hookinfo, 74
 - hook, 74
- etf_p
 - ng_etf.c, 1272
- ether
 - ng_atmllc_priv, 122
 - ng_eiface_private, 197
- ETHER_EQUAL
 - ng_bridge.c, 1216
- ethernet
 - ng_rfc1490_private, 382
- ethernet_hook
 - PPPoE, 463
- ethertype
 - ng_etffilter, 199
- ETHERTYPE_PPPOE_3COM_DISC
 - ng_pppoe.h, 1531
- ETHERTYPE_PPPOE_3COM_SESS
 - ng_pppoe.h, 1531
- ETHERTYPE_PPPOE_DISC
 - ng_pppoe.h, 1532
- ETHERTYPE_PPPOE_SESS
 - ng_pppoe.h, 1532
- event
 - __attribute__, 45
- event_mask
 - __attribute__, 45
- evnt_recv
 - ng_hci_node_stat_ep, 231
- exp_callout
 - netflow, 104
- expire_flow
 - netflow.c, 1093
- export
 - netflow, 104
- export_add
 - netflow.c, 1094
- export_item
 - netflow, 105
- export_mtx
 - netflow, 105

- export_send
 - netflow.c, 1094
- EXTRA
 - ngatmbase.c, 606
- f
 - flow_entry, 76
- failAlg
 - ng_one2many_config, 341
- family
 - iffam, 95
 - ng_ksocket_alias, 257
- fcs
 - ng_async_private, 118
 - rfcomm_cmd_hdr, 481
- fcstab
 - ng_async.c, 1162
- fddi
 - ng_atmllc_priv, 122
- features
 - __attribute__, 45
 - ng_btsocket_hci_raw_node_features, 152
 - ng_hci_neighbor, 222
 - ng_hci_node_neighbor_cache_entry_ep, 230
 - ng_hci_unit, 237
- FEC_BTTYPE_INET
 - ng_fec.c, 1290
- FEC_BTTYPE_INET6
 - ng_fec.c, 1290
- FEC_BTTYPE_MAC
 - ng_fec.c, 1291
- fec_bundle
 - ng_fec_private, 204
- FEC_BUNDLESIZ
 - ng_fec.c, 1291
- fec_ch
 - ng_fec_private, 204
- fec_idx
 - ng_fec_portlist, 203
- fec_if
 - ng_fec_portlist, 203
- fec_if_input
 - ng_fec_portlist, 203
- fec_ifstat
 - ng_fec_portlist, 203
- fec_mac
 - ng_fec_portlist, 203
- file
 - unimem_debug, 520
- filter, 75
 - ng_btsocket_hci_raw_pcb, 160
 - ng_vlan_table, 415
- filter_condition_type
 - __attribute__, 45
- filter_type
 - __attribute__, 45
- findhook
 - ng_type, 404
- first
 - flow_entry_data, 77
 - netflow_v1_record, 107
 - netflow_v5_record, 114
 - ng_ppp_frag, 355
- fixed
 - nglmistat, 423
- FixedPkts
 - ng_tcpmss_hookstat, 400
- flags
 - __attribute__, 45
 - bt3c_softc, 55
 - ctxinfo, 69
 - ETF, 72
 - frmrel_softc, 84
 - greheader, 85
 - netflow_v1_record, 107
 - netflow_v5_record, 114
 - ng_btsocket_hci_raw_pcb, 161
 - ng_btsocket_l2cap_pcb, 164
 - ng_btsocket_l2cap_raw_node_flags, 173
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_btsocket_rfcomm_pcb, 183
 - ng_btsocket_rfcomm_session, 186
 - ng_h4_info, 209
 - ng_hci_lp_qos_req_ep, 220
 - ng_hci_unit_con, 242
 - ng_l2cap, 262
 - ng_l2cap_cmd, 270
 - ng_l2cap_con, 273
 - ng_l2cap_node_con_ep, 306
 - ng_mesg::ng_msghdr, 322
 - ng_nat_priv, 328
 - ngd_private, 417
 - nglmi_softc, 420
 - ngm_atm_cpcs_init, 426
 - ngsock, 455
 - ngt_sc, 456
 - ngvcc, 458
 - PPPoE, 464
 - private, 475
 - XXX, 528
- fle_i_ifx
 - ng_netflow.h, 1110
- fle_o_ifx
 - flow_entry_data, 77
- FLG_DEBUG
 - ng_tty.c, 1647
- FLG_DIE
 - ng_tty.c, 1647

- flow
 - ng_l2cap_cfg_opt_val_t, 264
 - priv, 470
- flow_control
 - __attribute__, 45
 - rfcomm_mcc_pn, 485
 - rfcomm_mcc_rpn, 488
- flow_entry, 76
 - f, 76
 - TAILQ_ENTRY, 76
- flow_entry_data, 77
 - bytes, 77
 - dst_mask, 77
 - first, 77
 - file_o_ifx, 77
 - last, 78
 - next_hop, 78
 - packets, 78
 - r, 78
 - src_mask, 78
 - tcp_flags, 78
- flow_hash_entry, 79
 - mtx, 79
 - TAILQ_HEAD, 79
- flow_lower
 - ng_sscop.c, 624
- flow_manager, 80
 - id, 80
- flow_rec, 81
 - all, 81
 - both, 81
 - d_port, 81
 - dir, 81
 - i, 82
 - i_ifx, 82
 - misc, 82
 - ports, 82
 - prot, 82
 - r_dst, 82
 - r_src, 82
 - s_port, 82
 - tos, 82
- flow_seq
 - netflow, 105
 - netflow_v5_header, 111
- flow_upper
 - ng_sscop.c, 624
- flush_timo
 - ng_btsocket_l2cap_pcb, 164
 - ng_l2cap_cfg_opt_val_t, 264
 - ng_l2cap_chan, 266
 - ng_l2cap_l2ca_cfg_ind_ip, 277
 - ng_l2cap_l2ca_cfg_ip, 278
 - ng_l2cap_l2ca_cfg_op, 279
- flushed
 - ng_mppc_dir, 325
- fn
 - ng_item, 254
 - ng_ppp.c, 1506
- fn_arg1
 - ng_item, 254
- fn_arg2
 - ng_item, 254
- fn_fn
 - ng_item, 255
- frames
 - ng_hole_hookstat, 245
- FramesComp
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- FramesPlain
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- FramesUncomp
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- frmrel_softc, 83
 - addrlen, 83
 - ALT, 83
 - channel, 84
 - datahooks, 84
 - downstream, 84
 - flags, 84
 - local_seq, 84
 - mtu, 84
 - node, 84
 - remote_seq, 84
 - unit, 84
- ftarget
 - ng_l2tp_private, 311
- FULL_HASH
 - netflow.c, 1092
- fw_node
 - ng_ipfw.c, 1355
- get_export_dgram
 - netflow.c, 1094
- get_hook_from_iffam
 - ng_gif_demux.c, 1320
 - ng_iface.c, 1339
- get_iffam_from_af
 - ng_gif_demux.c, 1320
 - ng_iface.c, 1339
- get_iffam_from_hook
 - ng_gif_demux.c, 1320
 - ng_iface.c, 1339
- get_iffam_from_name
 - ng_gif_demux.c, 1320

- ng_iface.c, 1339
- get_new_sid
 - ng_pppoe.c, 1521
- get_next_l2cap_opt
 - ng_l2cap_evnt.c, 936
- get_tag
 - ng_pppoe.c, 1521
- getAlign
 - ng_parse_type, 350
- getDefault
 - ng_parse_array_info, 346
 - ng_parse_fixedarray_info, 347
 - ng_parse_type, 350
- getLength
 - ng_parse_array_info, 346
- gFamilies
 - ng_gif_demux.c, 1322
 - ng_iface.c, 1342
- gif
 - ng_gif_demux_private, 206
- got
 - ng_h4_info, 209
- greheader, 85
 - cid, 85
 - data, 85
 - flags, 85
 - hasAck, 85
 - hasKey, 85
 - hasRoute, 85
 - hasSeq, 86
 - hasSum, 86
 - length, 86
 - proto, 86
 - recursion, 86
 - ssr, 86
 - vers, 86
- GROUP4
 - ng_lmi.c, 1395
- GuessTable
 - ng_pred1_private, 376
- h2hc_flow
 - __attribute__, 45
- handle
 - cisco_priv, 67
 - nglmi_softc, 420
 - sess_neg, 494
- hardware_code
 - __attribute__, 46
- hardware_error
 - ng_hci_evnt.c, 733
- hasAck
 - greheader, 85
- HASH
 - ng_bridge.c, 1216
 - ng_etf.c, 1271
 - ng_pred1.c, 1555
 - ng_vlan.c, 1672
- Hash
 - ng_pred1_private, 376
- hash
 - netflow, 105
- hash_insert
 - netflow.c, 1094
- hashMask
 - ng_bridge_private, 139
- HASHSIZE
 - ng_etf.c, 1271
 - ng_vlan.c, 1672
- hashtable
 - ETF, 72
 - priv_p, 473
- hasKey
 - greheader, 85
- hasRoute
 - greheader, 85
- hasSeq
 - greheader, 86
- hasSum
 - greheader, 86
- hci
 - ng_l2cap, 262
- hci_family
 - sockaddr_hci, 496
- hci_len
 - sockaddr_hci, 496
- hci_node
 - sockaddr_hci, 496
- hci_revision
 - __attribute__, 46
- hci_version
 - __attribute__, 46
- HDLC_UI
 - ng_rfc1490.c, 1566
 - ng_UI.c, 1657
- hdr
 - datatag, 71
- head
 - ng_bt_itemq, 144
 - ng_bt_mbufq, 146
- HEAD_IS_READER
 - ng_base.c, 1179
- HEAD_IS_WRITER
 - ng_base.c, 1179
- header
 - netflow_v5_export_dgram, 110
 - ng_mesg, 320
- hi_match

- ng_tag_hookinfo, 395
 - hi_nonmatch
 - ng_tag_hookinfo, 395
 - high_watermark
 - ngm_queue_state, 441
 - hinfo_p
 - ng_bpf.c, 1206
 - ng_hole.c, 1327
 - ng_tag.c, 1623
 - HK_DEAD
 - netgraph.h, 1127
 - hk_flags
 - ng_hook, 247
 - HK_FORCE_WRITER
 - netgraph.h, 1127
 - HK_INVALID
 - netgraph.h, 1127
 - hk_name
 - ng_hook, 247
 - hk_node
 - ng_hook, 247
 - hk_peer
 - ng_hook, 247
 - hk_private
 - ng_hook, 247
 - HK_QUEUE
 - netgraph.h, 1127
 - hk_rcvdata
 - ng_hook, 247
 - hk_rcvmsg
 - ng_hook, 247
 - hk_refs
 - ng_hook, 247
 - hk_type
 - ng_hook, 247
 - hold_mode_activity
 - __attribute__, 46
 - hook
 - bt3c_softc, 55
 - cchhook, 59
 - ctxinfo, 69
 - ETF_hookinfo, 74
 - hookinfo, 87
 - ng_bpf_hookinfo, 125
 - ng_bridge_link, 134
 - ng_btsocket_l2cap_rtenry, 178
 - ng_h4_info, 209
 - ng_ipfw_hook_priv, 251
 - ng_ksocket_private, 258
 - ng_mppc_dir, 325
 - ng_netflow_iface, 330
 - ng_one2many_link, 342
 - ng_ppp_link, 357
 - ng_sppp_private, 388
 - ng_vatmpif_hook, 410
 - ng_vlan_filter, 414
 - ngd_private, 417
 - nglmistat, 423
 - ngpppoe_init_data, 451
 - ngpppoe_sts, 452
 - ngt_sc, 457
 - ngvcc, 458
 - protoent, 480
 - sess_con, 492
 - XXX_hookinfo, 529
 - hook_cnt
 - ccnode, 62
 - HOOK_INDEX_ATALK
 - ng_ppp.c, 1487
 - HOOK_INDEX_BYPASS
 - ng_ppp.c, 1487
 - HOOK_INDEX_COMPRESS
 - ng_ppp.c, 1487
 - HOOK_INDEX_DECOMPRESS
 - ng_ppp.c, 1487
 - HOOK_INDEX_DECRYPT
 - ng_ppp.c, 1487
 - HOOK_INDEX_ENCRYPT
 - ng_ppp.c, 1487
 - HOOK_INDEX_INET
 - ng_ppp.c, 1487
 - HOOK_INDEX_IPV6
 - ng_ppp.c, 1487
 - HOOK_INDEX_IPX
 - ng_ppp.c, 1487
 - HOOK_INDEX_MAX
 - ng_ppp.c, 1488
 - HOOK_INDEX_VJC_COMP
 - ng_ppp.c, 1488
 - HOOK_INDEX_VJC_IP
 - ng_ppp.c, 1488
 - HOOK_INDEX_VJC_UNCOMP
 - ng_ppp.c, 1488
 - HOOK_INDEX_VJC_VJIP
 - ng_ppp.c, 1488
- hook_p
 - netgraph.h, 1150
- hookinfo, 87
 - hook, 87
 - stats, 87
- hooklist, 88
 - link, 88
 - nodeinfo, 88
- hookname
 - iffam, 95
- hookpriv_p
 - ng_l2tp.c, 1379
- hooks

- ng_gif_demux_private, 206
- ng_iface_private, 249
- ng_ppp_private, 366
- nodeinfo, 461
- host
 - ng_bridge_hent, 131
- hosts
 - ng_bridge_host_ary, 133
- hotchar
 - ngt_sc, 457
- hpriv_p, 89
 - ng_ipfw.c, 1353
 - outHook, 89
 - stats, 89
- hva_st_aal5
 - vatmpif_stats, 523
- hva_st_atm
 - vatmpif_stats, 523
- hva_st_ng
 - vatmpif_stats, 523
- Hva_Stats_aal5
 - ng_atmpif.h, 591
- hva_stats_aal5, 90
 - aal5_crc_len, 90
 - aal5_drops, 90
 - aal5_pdu_crc, 90
 - aal5_pdu_drops, 90
 - aal5_pdu_errs, 90
 - aal5_pdu_rcvd, 90
 - aal5_pdu_xmit, 90
 - aal5_rcvd, 91
 - aal5_xmit, 91
- HVA_STATS_AAL5_NG_TYPE_INFO
 - ng_atmpif.h, 589
- Hva_Stats_atm
 - ng_atmpif.h, 591
- hva_stats_atm, 92
 - atm_rcvd, 92
 - atm_xmit, 92
- HVA_STATS_ATM_NG_TYPE_INFO
 - ng_atmpif.h, 589
- Hva_Stats_ng
 - ng_atmpif.h, 591
- hva_stats_ng, 93
 - ng_badpdu, 93
 - ng_errseq, 93
 - ng_lostpdu, 93
 - ng_rx_iqfull, 93
 - ng_rx_novcc, 93
 - ng_rx_pdu, 93
 - ng_rx_rawcell, 93
 - ng_tx_pdu, 94
 - ng_tx_rawcell, 94
- HVA_STATS_NG_TYPE_INFO
 - ng_atmpif.h, 589
- hwassist
 - private, 475
- i
 - flow_rec, 82
- i_ifx
 - flow_rec, 82
 - netflow_v5_record, 114
- ibuf
 - ng_h4_info, 209
- icr
 - ngm_atm_cpcs_init, 427
- id
 - flow_manager, 80
 - nodeinfo, 461
 - sa_tag, 490
- ident
 - __attribute__, 46
 - ng_l2cap_chan, 266
 - ng_l2cap_cmd, 270
 - ng_l2cap_con, 273
 - ng_l2cap_l2ca_con_ind_ip, 282
 - ng_l2cap_l2ca_con_rsp_ip, 285
- ierrors
 - ng_bt3c_node_stat_ep, 143
 - ng_h4_node_stat_ep, 211
 - ng_ubt_node_stat_ep, 407
- if_error
 - ng_fec_private, 205
- if_flags
 - ng_fec_private, 205
- iface
 - ng_netflow_setdlt, 335
 - ng_netflow_setifindex, 336
- iface_p
 - ng_netflow.h, 1113
- ifaces
 - netflow, 105
- iffam, 95
 - family, 95
 - hookname, 95
- iffam_p
 - ng_gif_demux.c, 1320
 - ng_iface.c, 1339
- IFFLAGS
 - ng_atm.c, 570
- ifinfo_dlt
 - ng_netflow_ifinfo, 332
- ifinfo_index
 - ng_netflow_ifinfo, 332
- ifinfo_p
 - ng_netflow.h, 1113
- ifinfo_packets

- ng_netflow_ifinfo, 332
- iflow
 - ng_btsocket_l2cap_pcb, 165
 - ng_l2cap_chan, 266
 - ng_l2cap_l2ca_cfg_ind_ip, 277
 - ng_l2cap_l2ca_cfg_rsp_ip, 280
- ifMatch
 - ng_bpf_hookprog, 127
 - ng_tag_hookin, 393
- ifmedia
 - ng_fec_private, 205
- ifNotMatch
 - ng_bpf_hookprog, 127
 - ng_tag_hookin, 393
- ifp
 - ng_eiface_private, 197
 - ng_fec_private, 205
 - ng_iface_private, 249
 - ng_ipfw_tag, 252
 - ng_sppp_private, 388
 - priv, 470
 - private, 476
- IFP2NG
 - ng_atm.c, 570
 - ng_ether.c, 1280
 - ng_fec.c, 1291
 - ng_gif.c, 1310
- IFP2NG_SET
 - ng_atm.c, 570
 - ng_gif.c, 1310
- ifqmaxlen
 - ng_btsocket_hci_raw.c, 1026
 - ng_btsocket_l2cap.c, 1048
 - ng_btsocket_l2cap_raw.c, 1060
 - ng_btsocket_rfcomm.c, 1088
- imtu
 - ng_btsocket_l2cap_pcb, 165
 - ng_l2cap_chan, 267
 - ng_l2cap_l2ca_cfg_ip, 278
 - ng_l2cap_l2ca_cfg_op, 279
 - ng_l2cap_node_chan_ep, 303
- in
 - ng_nat_priv, 328
 - ng_split_private, 386
 - ng_tag_hookinfo, 395
- in_dropped
 - stats, 505
- in_errors
 - ngm_atm_stats, 431
 - priv, 470
- in_ifx
 - netflow_v1_record, 107
- in_packets
 - ngm_atm_stats, 431
 - priv, 470
 - stats, 505
- in_tag_cookie
 - ng_tag_hookinfo, 396
- in_tag_data
 - ng_tag_hookinfo, 396
- in_tag_id
 - ng_tag_hookinfo, 396
- in_tag_len
 - ng_tag_hookinfo, 396
- INACTIVE
 - netflow.c, 1092
- INACTIVE_TIMEOUT
 - netflow.h, 1098
- inactive_timeout
 - ng_netflow_settimeouts, 337
- inbuf
 - ng_deflate_private, 194
 - ng_pred1_private, 376
- include_length
 - ng_l2tp_sess_config, 314
- index
 - ng_netflow_setifindex, 336
- inet
 - cisco_priv, 67
 - ng_rfc1490_private, 382
- inet6
 - cisco_priv, 67
- info
 - netflow, 105
 - ng_netflow_iface, 330
 - ng_parse_type, 350
- info_data
 - ng_btsocket_l2cap_raw_get_info, 171
- info_size
 - ng_btsocket_l2cap_raw_get_info, 171
 - ng_l2cap_l2ca_get_info_op, 291
- info_type
 - ng_btsocket_l2cap_raw_get_info, 171
 - ng_l2cap_l2ca_get_info_ip, 290
- inFrames
 - ng_tee_hookstat, 401
- inHook
 - ng_tcpmss_config, 399
- init_tags
 - ng_pppoe.c, 1521
- INITIATOR
 - ng_btsocket_rfcomm.h, 826
- InOctets
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- inOctets
 - ng_tee_hookstat, 401
- input

- priv, 470
- privdata, 478
- inq
 - bt3c_softc, 55
- inquiry_length
 - __attribute__, 46
- inquiry_result
 - ng_hci_evnt.c, 733
- inquiry_scan_interval
 - __attribute__, 46
- inquiry_scan_window
 - __attribute__, 46
- INS_ALLOC
 - ng_uni_cust.h, 653
- INS_FREE
 - ng_uni_cust.h, 653
- InSeq
 - ng_atmpif_link_status, 124
 - ng_vatmpif_hook, 410
- insert_tag
 - ng_pppoe.c, 1521
- inst
 - cchook, 59
- INT16_ALIGNMENT
 - ng_parse.c, 1451
- int16_temp, 96
 - x, 96
 - y, 96
- INT32_ALIGNMENT
 - ng_parse.c, 1451
- int32_temp, 97
 - x, 97
 - y, 97
- INT64_ALIGNMENT
 - ng_parse.c, 1451
- int64_temp, 98
 - x, 98
 - y, 98
- INT8_ALIGNMENT
 - ng_parse.c, 1451
- INT_HEX
 - ng_parse.c, 1451
- INT_SIGNED
 - ng_parse.c, 1451
- INT_UNSIGNED
 - ng_parse.c, 1451
- interval
 - __attribute__, 46
- intr_ch
 - privdata, 478
- invalidx
 - nglmi_softc, 420
- INVOKE
 - ng_parse.c, 1452
- iobase
 - bt3c_softc, 55
- iobase_rid
 - bt3c_softc, 55
- ioh
 - bt3c_softc, 55
- iot
 - bt3c_softc, 56
- ip
 - ng_vjc_private, 412
- ip_hash
 - netflow.c, 1095
- ipaddr
 - ng_cisco_ipaddr, 189
- ipfw
 - ng_bridge_config, 130
- ipx
 - cisco_priv, 67
- irq
 - bt3c_softc, 56
- irq_cookie
 - bt3c_softc, 56
- irq_rid
 - bt3c_softc, 56
- is_uni
 - cchook, 60
- IS_VATMPIF_DEBUG_PACKET
 - ng_atmpif.h, 589
- ISFREE
 - netflow.c, 1093
- ITEM_DEBUG_CHECKS
 - ng_base.c, 1179
- item_p
 - netgraph.h, 1150
- ith
 - bt3c_softc, 56
- KEEPALIVE_SECS
 - ng_cisco.c, 1231
- keepAlivePeriod
 - ng_cisco_stats, 190
- key
 - __attribute__, 46
- key_flag
 - __attribute__, 46
- key_type
 - __attribute__, 47
- KEYLEN
 - ng_mppc.c, 1421
- KSF_ACCEPTING
 - ng_ksocket.c, 1362
- KSF_CLONED
 - ng_ksocket.c, 1362
- KSF_CONNECTING

- ng_ksocket.c, [1362](#)
- KSF_EMBRYONIC
 - ng_ksocket.c, [1362](#)
- KSF_EOFSEEN
 - ng_ksocket.c, [1362](#)
- l2c
 - ng_l2cap, [263](#)
- l2cap
 - ng_l2cap_con, [273](#)
- l2cap_bdaddr
 - sockaddr_l2cap, [497](#)
- l2cap_family
 - sockaddr_l2cap, [497](#)
- l2cap_len
 - sockaddr_l2cap, [497](#)
- l2cap_psm
 - sockaddr_l2cap, [497](#)
- l2so
 - ng_btsocket_rfcomm_session, [186](#)
- L2TP_CONTROL_DSEQ
 - ng_l2tp.c, [1376](#)
- L2TP_COPY_MBUF
 - ng_l2tp.c, [1376](#)
- L2TP_CTRL_0BITS
 - ng_l2tp.c, [1376](#)
- L2TP_CTRL_1BITS
 - ng_l2tp.c, [1377](#)
- L2TP_CTRL_HDR
 - ng_l2tp.c, [1377](#)
- L2TP_DATA_0BITS
 - ng_l2tp.c, [1377](#)
- L2TP_DATA_1BITS
 - ng_l2tp.c, [1377](#)
- L2TP_DATA_HDR
 - ng_l2tp.c, [1377](#)
- L2TP_DELAYED_ACK
 - ng_l2tp.c, [1377](#)
- L2TP_ENABLE_DSEQ
 - ng_l2tp.c, [1377](#)
- L2TP_HDR_CTRL
 - ng_l2tp.c, [1377](#)
- L2TP_HDR_LEN
 - ng_l2tp.c, [1378](#)
- L2TP_HDR_OFF
 - ng_l2tp.c, [1378](#)
- L2TP_HDR_PRIO
 - ng_l2tp.c, [1378](#)
- L2TP_HDR_SEQ
 - ng_l2tp.c, [1378](#)
- L2TP_HDR_VERS_MASK
 - ng_l2tp.c, [1378](#)
- L2TP_HDR_VERSION
 - ng_l2tp.c, [1378](#)
- L2TP_MAX_REXMIT
 - ng_l2tp.c, [1378](#)
- L2TP_MAX_REXMIT_TO
 - ng_l2tp.c, [1378](#)
- L2TP_MAX_XWIN
 - ng_l2tp.c, [1378](#)
- l2tp_seq, [99](#)
 - acks, [99](#)
 - cwnd, [99](#)
 - max_rexmit_to, [99](#)
 - max_rexmits, [99](#)
 - nr, [99](#)
 - ns, [100](#)
 - rack, [100](#)
 - rack_timer, [100](#)
 - rexmits, [100](#)
 - ssth, [100](#)
 - wmax, [100](#)
 - xack, [100](#)
 - xack_timer, [100](#)
 - xwin, [101](#)
- L2TP_SEQ_CHECK
 - ng_l2tp.c, [1379](#)
- L2TP_SEQ_DIFF
 - ng_l2tp.c, [1379](#)
- lap
 - __attribute__, [47](#)
- last
 - flow_entry_data, [78](#)
 - netflow_v1_record, [108](#)
 - netflow_v5_record, [114](#)
 - ng_ppp_frag, [355](#)
 - ng_queue, [379](#)
 - ngnf_flows, [448](#)
- LASTBYTE_D_C
 - ng_frame_relay.c, [1302](#)
- lastLink
 - ng_ppp_private, [366](#)
- lastTime
 - ng_source_stats, [384](#)
- lasttime
 - ng_async_private, [118](#)
- lastWrite
 - ng_ppp_link, [358](#)
- latency
 - __attribute__, [47](#)
 - ng_hci_lp_qos_req_ep, [220](#)
 - ng_ppp_link, [358](#)
 - ng_ppp_link_conf, [359](#)
- lbmode
 - __attribute__, [47](#)
- lcid
 - __attribute__, [47](#)
 - ng_l2cap_l2ca_cfg_ind_ip, [277](#)

- ng_l2cap_l2ca_cfg_ip, 278
- ng_l2cap_l2ca_cfg_rsp_ip, 280
- ng_l2cap_l2ca_con_ind_ip, 282
- ng_l2cap_l2ca_con_op, 284
- ng_l2cap_l2ca_con_rsp_ip, 285
- ng_l2cap_l2ca_discon_ip, 287
- ng_l2cap_l2ca_grp_add_member_ip, 292
- ng_l2cap_l2ca_grp_close_ip, 294
- ng_l2cap_l2ca_grp_create_op, 296
- ng_l2cap_l2ca_grp_get_members_ip, 297
- ng_l2cap_l2ca_write_op, 302
- LEAVE
- ng_pppoe.c, 1519
- left
 - ng_tee_stats, 402
 - privdata, 478
- left2right
 - ng_tee_stats, 402
 - privdata, 478
- len
 - ng_bt_itemq, 144
 - ng_bt_mbufq, 146
- length
 - __attribute__, 47
 - greheader, 86
 - ng_l2cap_l2ca_write_op, 302
 - ng_parse_fixedarray_info, 347
 - pppoe_hdr, 466
 - rfcomm_cmd_hdr, 481
 - rfcomm_frame_hdr, 482
 - rfcomm_mcc_hdr, 483
- level
 - __attribute__, 47
 - ng_ksocket_sockopt, 260
 - ngm_uni_debug, 445
- lib
 - ng_nat_priv, 328
- line_settings
 - rfcomm_mcc_rpn, 488
- link
 - hooklist, 88
 - vatmpif_unit, 524
- LINK_NUM
 - ng_bridge.c, 1216
- link_policy_mask
 - ng_hci_unit, 237
- link_timo
 - ng_btsocket_l2cap_pcb, 165
 - ng_l2cap_chan, 267
 - ng_l2cap_l2ca_cfg_ip, 278
- link_type
 - __attribute__, 47
 - ng_hci_lp_con_cfm_ep, 212
 - ng_hci_lp_con_ind_ep, 213
 - ng_hci_lp_con_req_ep, 214
 - ng_hci_lp_con_rsp_ep, 215
 - ng_hci_lp_discon_ind_ep, 216
 - ng_hci_node_con_ep, 226
 - ng_hci_unit_con, 242
- linkinfo, 102
 - nodeinfo, 102
 - ourhook, 102
 - peerhook, 102
- linkNum
 - ng_bridge_host, 132
- links
 - ng_bridge_private, 140
 - ng_ppp_node_conf, 364
 - ng_ppp_private, 366
- LIST_ENTRY
 - ng_btsocket_hci_raw_pcb, 160
 - ng_btsocket_l2cap_pcb, 164
 - ng_btsocket_l2cap_raw_pcb, 174
 - ng_btsocket_l2cap_rentry, 178
 - ng_btsocket_rfcomm_pcb, 183
 - ng_btsocket_rfcomm_session, 186
 - ng_hci_neighbor, 222
 - ng_hci_unit_con, 242
 - ng_hook, 247
 - ng_l2cap_chan, 266
 - ng_l2cap_con, 273
 - ng_node, 339
 - ng_type, 404
 - ngpcb, 450
- LIST_HEAD
 - ng_base.c, 1183
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap_raw.c, 1055
 - ng_btsocket_rfcomm.c, 1068
 - ng_btsocket_rfcomm_session, 186
 - ng_etf.c, 1272
 - ng_hci_unit, 236
 - ng_l2cap, 262
 - ng_node, 339
 - ng_uni.c, 642
 - ng_vlan.c, 1673
 - ngatmbase.c, 606
- liv_per_full
 - nglmi_softc, 420
- liv_rate
 - nglmi_softc, 420
- lives
 - nglmi_softc, 420
- lmi_annexA
 - nglmi_softc, 420
- lmi_annexD
 - nglmi_softc, 420

- lmi_channel
 - nglmi_softc, 420
- lmi_channel0
 - nglmi_softc, 421
- lmi_channel1023
 - nglmi_softc, 421
- lmi_group4
 - nglmi_softc, 421
- LMI_MIN_LENGTH
 - ng_lmi.c, 1395
- LMI_PATIENCE
 - ng_lmi.c, 1395
- LMI_ticker
 - ng_lmi.c, 1398
- LMIPOLLSIZE
 - ng_lmi.c, 1395
- lmodem
 - ng_btsocket_rfcomm_fc_info, 180
 - ng_btsocket_rfcomm_pcb, 183
- lmp_max_slots
 - __attribute__, 47
- lmp_subversion
 - __attribute__, 47
- lmp_version
 - __attribute__, 48
- lno
 - unimem_debug, 520
- local_seq
 - cisco_priv, 67
 - firmrel_softc, 84
 - nglmi_softc, 421
- localip
 - cisco_priv, 67
- localmask
 - cisco_priv, 67
- loopCount
 - ng_bridge_link, 134
- loopDetects
 - ng_bridge_link_stats, 136
- loopDrops
 - ng_bridge_link_stats, 136
- loopTimeout
 - ng_bridge_config, 130
- low_watermark
 - ngm_queue_state, 441
- lower
 - ng_l2tp_private, 311
 - ng_pptpgre_private, 371
 - priv, 470
 - private, 476
- lowerOrphan
 - private, 476
- m
 - bt3c_softc, 56
 - ngt_sc, 457
 - sess_neg, 495
 - M_CHECK
 - ng_netflow.c, 1102
 - ng_tcpmss.c, 1631
 - M_FEC_INET
 - ng_fec.h, 1297
 - M_FEC_INET6
 - ng_fec.h, 1297
 - M_FEC_MAC
 - ng_fec.h, 1297
 - M_NETGRAPH_ASYNC
 - ng_async.c, 1160
 - M_NETGRAPH_ATMPIF
 - ng_atmpif.c, 535
 - M_NETGRAPH_BPF
 - ng_bpf.c, 1205
 - M_NETGRAPH_BRIDGE
 - ng_bridge.c, 1216
 - M_NETGRAPH_BT_SOCKET_HCI_RAW
 - ng_btsocket_hci_raw.c, 1019
 - M_NETGRAPH_BT_SOCKET_L2CAP
 - ng_btsocket_l2cap.c, 1032
 - M_NETGRAPH_BT_SOCKET_L2CAP_RAW
 - ng_btsocket_l2cap_raw.c, 1054
 - M_NETGRAPH_BT_SOCKET_RFCOMM
 - ng_btsocket_rfcomm.c, 1066
 - M_NETGRAPH ETF
 - ng_etf.c, 1272
 - M_NETGRAPH_GIF_DEMUX
 - ng_gif_demux.c, 1319
 - M_NETGRAPH_HCI
 - ng_hci_main.c, 743
 - ng_hci_var.h, 777
 - M_NETGRAPH_IFACE
 - ng_iface.c, 1338
 - M_NETGRAPH_KSOCKET
 - ng_ksocket.c, 1362
 - M_NETGRAPH_L2CAP
 - ng_l2cap_main.c, 961
 - ng_l2cap_var.h, 1007
 - M_NETGRAPH_L2TP
 - ng_l2tp.c, 1379
 - M_NETGRAPH_MPPC
 - ng_mppc.c, 1421
 - M_NETGRAPH_PARSE
 - ng_parse.c, 1452
 - M_NETGRAPH_PATH
 - ng_socket.c, 1583
 - M_NETGRAPH_PPP
 - ng_ppp.c, 1488
 - M_NETGRAPH_PPPOE
 - ng_pppoe.c, 1519

- M_NETGRAPH_SOCKET
 - ng_socket.c, 1583
- M_NETGRAPH_SPPP
 - ng_sppp.c, 1614
- M_NETGRAPH_TAG
 - ng_tag.c, 1622
- M_NETGRAPH_XXX
 - ng_sample.c, 1574
- maa_dropped
 - stats, 506
- maa_signals
 - stats, 506
- macaddr
 - ng_vatmpif_config, 409
- make_packet
 - ng_pppoe.c, 1521
- makeup
 - ng_frame_relay.c, 1305
- MALLOC_DECLARE
 - netflow.c, 1095
 - netgraph.h, 1151
 - ng_ccatm_cust.h, 566
 - ng_h4_var.h, 690
- MALLOC_DEFINE
 - netflow.c, 1095
 - ng_bt3c_pccard.c, 669
 - ng_ccatm.c, 555
 - ng_deflate.c, 1242
 - ng_h4.c, 680
 - ng_pred1.c, 1556
 - ng_sscfu.c, 613
 - ng_sscop.c, 624
 - ng_uni.c, 643
 - ngatmbase.c, 606, 607
- manage
 - ccnode, 62
 - priv, 470
- manufacturer
 - __attribute__, 48
- many
 - ng_one2many_private, 344
- mask
 - ng_sscfu_getdefparam, 390
 - ng_sscop_setparam, 391
 - ng_sscop_setparam_resp, 392
 - ngm_uni_config_mask, 444
 - ngm_uni_set_config, 446
 - segment, 491
- match_id
 - ng_l2tp_config, 308
- matchhook
 - ng_etfilter, 199
- max_acl_size
 - __attribute__, 48
- MAX_BUCKETS
 - ng_bridge.c, 1217
- MAX_CT
 - ng_frame_relay.c, 1302
- MAX_DLCIS
 - ng_lmi.c, 1395
- MAX_ENCAPS_HDR
 - ng_rfc1490.c, 1566
 - ng_UI.c, 1657
- max_interval
 - __attribute__, 48
- MAX_MBUFQ
 - ng_tty.c, 1648
- MAX_NGD
 - ng_device.c, 1251
- max_num_keys
 - __attribute__, 48
- max_period_length
 - __attribute__, 48
- max_queuelen_bytes
 - ngm_queue_state, 441
- max_queuelen_packets
 - ngm_queue_state, 441
- max_retrans
 - rfcomm_mcc_pn, 485
- max_rexmit_to
 - l2tp_seq, 99
- max_rexmits
 - l2tp_seq, 99
- max_sco_size
 - __attribute__, 48
- max_vccs
 - ngm_atm_config, 425
- MAX_VJHEADER
 - ng_vjc.c, 1663
- max_vpcs
 - ngm_atm_config, 425
- maxalloc
 - ng_base.c, 1200
- maxChannel
 - ngm_vjc_config, 447
- MAXDLCI
 - ng_lmi.c, 1396
- MAXDLTNAMELEN
 - ng_netflow.h, 1110
- maxlen
 - ng_bt_itemq, 145
 - ng_bt_mbufq, 146
- maxMSS
 - ng_tcpmss_config, 399
 - ng_tcpmss_hookstat, 400
- maxPps
 - ng_source_stats, 384
- maxStaleness

- ng_bridge_config, 130
- mbs
 - ngm_atm_cpcs_init, 427
- MBUF_ALLOC
 - ng_sscop_cust.h, 633
- MBUF_APPEND32
 - ng_sscop_cust.h, 633
- MBUF_DUP
 - ng_sscop_cust.h, 633
- MBUF_FREE
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 633
- MBUF_GET32
 - ng_sscop_cust.h, 633
- MBUF_LEN
 - ng_sscop_cust.h, 633
- MBUF_PAD4
 - ng_sscop_cust.h, 633
- MBUF_STRIP32
 - ng_sscop_cust.h, 634
- MBUF_TRAIL32
 - ng_sscop_cust.h, 634
- MBUF_UNPAD
 - ng_sscop_cust.h, 634
- mcr
 - ngm_atm_cpcs_init, 427
- media
 - ng_atm.c, 576
- MEMFREE
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 634
- MEMINIT
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 634
- memmove
 - ng_l2tp.c, 1379
 - ng_uni_cust.h, 653
- memoryFailures
 - ng_bridge_link_stats, 136
 - ng_l2tp_stats, 316
 - ng_one2many_link_stats, 343
 - ng_pptpgre_stats, 373
- MEMZALLOC
 - ng_sscfu_cust.h, 619
 - ng_sscop_cust.h, 634
- msgType
 - ng_cmdlist, 191
- meta_p
 - netgraph.h, 1150
- METHOD
 - ng_parse.c, 1452
- method
 - ng_rfc1490_encap_t, 381
- MILLIuptime
 - netflow.c, 1093
- min
 - ng_hci_cmds.c, 720
- MIN_BUCKETS
 - ng_bridge.c, 1217
- min_interval
 - __attribute__, 48
- min_period_length
 - __attribute__, 48
- minStableAge
 - ng_bridge_config, 130
- misc
 - flow_rec, 82
- mixed
 - ng_split_private, 386
- mod_event
 - ng_type, 404
- mode
 - ng_hci_node_con_ep, 226
 - ng_hci_unit_con, 243
- mode_change
 - ng_hci_evnt.c, 733
- MODE_ESC
 - ng_async.c, 1160
- MODE_HUNT
 - ng_async.c, 1160
- MODE_NORMAL
 - ng_async.c, 1160
- modem
 - rfcomm_mcc_msc, 484
- MODULE_DEPEND
 - ng_bt3c_pccard.c, 669
 - ng_btsocket.c, 1012
 - ng_ccatm.c, 555
 - ng_deflate.c, 1242
 - ng_gif.c, 1311
 - ng_hci_main.c, 743
 - ng_ipfw.c, 1353
 - ng_l2cap_main.c, 961
 - ng_nat.c, 1432
 - ng_sppp.c, 1615
 - ng_sscfu.c, 613
 - ng_sscop.c, 624
 - ng_ubt.c, 694
 - ng_uni.c, 643
- MODULE_VERSION
 - ng_base.c, 1183
 - ng_bluetooth.c, 660
 - ng_bt3c_pccard.c, 669
 - ng_btsocket.c, 1012
 - ng_h4.c, 680
 - ng_hci_main.c, 743
 - ng_l2cap_main.c, 961
 - ng_ubt.c, 694

- ngatmbase.c, 607
- MP_FRAGTIMER_INTERVAL
 - ng_ppp.c, 1488
- MP_INITIAL_SEQ
 - ng_ppp.c, 1488
- MP_LONG_EXTEND
 - ng_ppp.c, 1488
- MP_LONG_FIRST_FLAG
 - ng_ppp.c, 1489
- MP_LONG_LAST_FLAG
 - ng_ppp.c, 1489
- MP_LONG_SEQ_DIFF
 - ng_ppp.c, 1489
- MP_LONG_SEQ_HIBIT
 - ng_ppp.c, 1489
- MP_LONG_SEQ_MASK
 - ng_ppp.c, 1489
- MP_MAX_QUEUE_LEN
 - ng_ppp.c, 1489
- MP_MIN_FRAG_LEN
 - ng_ppp.c, 1489
- MP_MIN_LINK_MRU
 - ng_ppp.c, 1489
- MP_MIN_MRRU
 - ng_ppp.c, 1489
- MP_NEXT_RECV_SEQ
 - ng_ppp.c, 1490
- MP_NOSEQ
 - ng_ppp.c, 1490
- MP_RECV_SEQ_DIFF
 - ng_ppp.c, 1490
- MP_SHORT_EXTEND
 - ng_ppp.c, 1490
- MP_SHORT_FIRST_FLAG
 - ng_ppp.c, 1490
- MP_SHORT_LAST_FLAG
 - ng_ppp.c, 1490
- MP_SHORT_SEQ_DIFF
 - ng_ppp.c, 1491
- MP_SHORT_SEQ_HIBIT
 - ng_ppp.c, 1491
- MP_SHORT_SEQ_MASK
 - ng_ppp.c, 1491
- MPPC_BIT
 - ng_mppc.h, 1427
- MPPC_CCOUNT_MASK
 - ng_mppc.c, 1421
- MPPC_COMP_OK
 - ng_mppc.c, 1421
- MPPC_DECOMP_BUFSIZE
 - ng_mppc.c, 1421
- MPPC_DECOMP_OK
 - ng_mppc.c, 1421
- MPPC_DECOMP_SAFETY
 - ng_mppc.c, 1421
- MPPC_FLAG_COMPRESSED
 - ng_mppc.c, 1421
- MPPC_FLAG_ENCRYPTED
 - ng_mppc.c, 1422
- MPPC_FLAG_FLUSHED
 - ng_mppc.c, 1422
- MPPC_FLAG_RESTART
 - ng_mppc.c, 1422
- MPPC_HDRLEN
 - ng_mppc.c, 1422
- MPPC_MAX_BLOWUP
 - ng_mppc.h, 1427
- MPPC_VALID_BITS
 - ng_mppc.h, 1427
- MPPE_128
 - ng_mppc.h, 1428
- MPPE_40
 - ng_mppc.h, 1428
- MPPE_56
 - ng_mppc.h, 1428
- MPPE_BITS
 - ng_mppc.h, 1428
- MPPE_KEY_LEN
 - ng_mppc.h, 1428
- MPPE_MAX_REKEY
 - ng_mppc.c, 1422
- MPPE_STATELESS
 - ng_mppc.h, 1428
- MPPE_UPDATE_FLAG
 - ng_mppc.c, 1422
- MPPE_UPDATE_MASK
 - ng_mppc.c, 1422
- mrru
 - ng_ppp_bund_conf, 353
- mru
 - ng_ppp_link_conf, 359
- mseq
 - ng_ppp_mp_state, 363
 - ng_ppp_private, 366
- msg
 - ng_btsocket_hci_raw_pcb, 161
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_item, 255
- MSG_ALLOC
 - ng_sscop_cust.h, 634
- MSG_FREE
 - ng_sscop_cust.h, 634
- msg_msg
 - ng_item, 255
- msg_retaddr
 - ng_item, 255
- MSGQ_APPEND
 - ng_sscop_cust.h, 634

- MSGQ_CLEAR
 - ng_sscop_cust.h, 634
- MSGQ_EMPTY
 - ng_sscop_cust.h, 635
- MSGQ_FOREACH
 - ng_sscop_cust.h, 635
- MSGQ_GET
 - ng_sscop_cust.h, 635
- MSGQ_INIT
 - ng_sscop_cust.h, 635
- MSGQ_INSERT_BEFORE
 - ng_sscop_cust.h, 635
- MSGQ_PEEK
 - ng_sscop_cust.h, 635
- MSGQ_REMOVE
 - ng_sscop_cust.h, 635
- mt
 - ng_ipfw_tag, 252
- mtu
 - firmrel_softc, 84
 - ng_btsocket_rfcomm_pcb, 183
 - ng_btsocket_rfcomm_session, 187
 - ng_l2cap_cfg_opt_val_t, 264
 - ng_l2cap_cmd_rej_data_t, 271
 - ng_l2cap_info_rsp_data_t, 276
 - rfcomm_mcc_pn, 485
- mtx
 - flow_hash_entry, 79
 - ng_pptpgre_private, 372
 - ngsock, 455
- n
 - ng_vlan_table, 415
- name
 - __attribute__, 48
 - ng_atm.c, 576
 - ng_cmdlist, 191
 - ng_ksocket_alias, 257
 - ng_ksocket_sockopt, 260
 - ng_parse_struct_field, 349
 - ng_ppp.c, 1506
 - ng_rfc1490_encap_t, 381
 - ng_type, 404
 - ngm_atm_cpcs_init, 427
 - ngm_atm_cpcs_term, 429
 - ngm_name, 440
 - nodeinfo, 461
- NAME_ANNEXA
 - ng_lmi.c, 1396
- NAME_ANNEXD
 - ng_lmi.c, 1396
- NAME_GROUP4
 - ng_lmi.c, 1396
- NAME_NONE
 - ng_lmi.c, 1396
- namelist, 103
 - nodeinfo, 103
 - numnames, 103
- names
 - ng_btsocket_hci_raw_node_list_names, 154
- NBUCKETS
 - netflow.c, 1093
- nd_flags
 - ng_node, 339
- nd_ID
 - ng_node, 339
- nd_input_queue
 - ng_node, 339
- nd_name
 - ng_node, 339
- nd_numhooks
 - ng_node, 339
- nd_private
 - ng_node, 339
- nd_refs
 - ng_node, 339
- nd_type
 - ng_node, 340
- need_full
 - nglmi_softc, 421
- neg
 - sess_con, 493
- negp
 - ng_pppoe.c, 1520
- nent
 - priv_p, 473
- nentries
 - ngnf_flows, 448
- NET_NEEDS_GIANT
 - ng_h4.c, 680
- netflow, 104
 - exp_callout, 104
 - export, 104
 - export_item, 105
 - export_mtx, 105
 - flow_seq, 105
 - hash, 105
 - ifaces, 105
 - info, 105
 - node, 105
 - zone, 105
- netflow.c
 - ADDR_HASH, 1092
 - AGED, 1092
 - expire_flow, 1093
 - export_add, 1094
 - export_send, 1094
 - FULL_HASH, 1092

- get_export_dgram, 1094
- hash_insert, 1094
- INACTIVE, 1092
- ip_hash, 1095
- ISFREE, 1093
- MALLOC_DECLARE, 1095
- MALLOC_DEFINE, 1095
- MILLIUPTIME, 1093
- NBUCKETS, 1093
- ng_netflow_cache_flush, 1095
- ng_netflow_cache_init, 1095
- ng_netflow_copyinfo, 1095
- ng_netflow_expire, 1096
- ng_netflow_flow_add, 1096
- ng_netflow_flow_show, 1096
- rcs_id, 1097
- return_export_dgram, 1096
- SMALL, 1093
- uma_ctor_flow, 1097
- uma_dtor_flow, 1097
- netflow.h
 - __packed__, 1099
 - ACTIVE_TIMEOUT, 1098
 - INACTIVE_TIMEOUT, 1098
 - NETFLOW_V1, 1098
 - NETFLOW_V1_MAX_RECORDS, 1099
 - NETFLOW_V1_MAX_SIZE, 1099
 - NETFLOW_V5, 1099
 - NETFLOW_V5_MAX_RECORDS, 1099
 - NETFLOW_V5_MAX_SIZE, 1099
- NETFLOW_V1
 - netflow.h, 1098
- netflow_v1_header, 106
 - count, 106
 - sys_uptime, 106
 - unix_nsecs, 106
 - unix_secs, 106
 - version, 106
- NETFLOW_V1_MAX_RECORDS
 - netflow.h, 1099
- NETFLOW_V1_MAX_SIZE
 - netflow.h, 1099
- netflow_v1_record, 107
 - d_port, 107
 - dst_addr, 107
 - first, 107
 - flags, 107
 - in_ifx, 107
 - last, 108
 - next_hop, 108
 - octets, 108
 - out_ifx, 108
 - packets, 108
 - pad1, 108
 - pad2, 108
 - pad3, 108
 - prot, 108
 - reserved, 108
 - s_port, 108
 - src_addr, 109
 - tos, 109
- NETFLOW_V5
 - netflow.h, 1099
- netflow_v5_export_dgram, 110
 - header, 110
 - r, 110
- netflow_v5_header, 111
 - count, 111
 - engine_id, 111
 - engine_type, 111
 - flow_seq, 111
 - pad, 111
 - sys_uptime, 111
 - unix_nsecs, 112
 - unix_secs, 112
 - version, 112
- NETFLOW_V5_MAX_RECORDS
 - netflow.h, 1099
- NETFLOW_V5_MAX_SIZE
 - netflow.h, 1099
- netflow_v5_record, 113
 - d_port, 113
 - dst_addr, 113
 - dst_as, 113
 - dst_mask, 113
 - first, 114
 - flags, 114
 - i_ifx, 114
 - last, 114
 - next_hop, 114
 - o_ifx, 114
 - octets, 114
 - packets, 114
 - pad1, 114
 - pad2, 115
 - prot, 115
 - s_port, 115
 - src_addr, 115
 - src_as, 115
 - src_mask, 115
 - tos, 115
- netgraph.h
 - _NGI_ARG1, 1125
 - _NGI_ARG2, 1125
 - _NGI_CLR_HOOK, 1125
 - _NGI_CLR_NODE, 1125
 - _NGI_FN, 1126
 - _NGI_HOOK, 1126

- [_NGI_M, 1126](#)
- [_NGI_META, 1126](#)
- [_NGI_MSG, 1126](#)
- [_NGI_NODE, 1126](#)
- [_NGI_RETADDR, 1126](#)
- [_NGI_SET_HOOK, 1126](#)
- [_NGI_SET_NODE, 1127](#)
- [_NG_ABI_VERSION, 1122](#)
- [_NG_HOOK_FORCE_QUEUE, 1122](#)
- [_NG_HOOK_FORCE_WRITER, 1122](#)
- [_NG_HOOK_IS_VALID, 1122](#)
- [_NG_HOOK_NAME, 1122](#)
- [_NG_HOOK_NODE, 1123](#)
- [_NG_HOOK_NOT_VALID, 1123](#)
- [_NG_HOOK_PEER, 1123](#)
- [_NG_HOOK_PRIVATE, 1123](#)
- [_NG_HOOK_REF, 1123](#)
- [_NG_HOOK_SET_PRIVATE, 1123](#)
- [_NG_HOOK_SET_RCVDATA, 1123](#)
- [_NG_HOOK_SET_RCVMSG, 1123](#)
- [_NG_HOOK_UNREF, 1123](#)
- [_NG_NODE_FORCE_WRITER, 1123](#)
- [_NG_NODE_FOREACH_HOOK, 1124](#)
- [_NG_NODE_HAS_NAME, 1124](#)
- [_NG_NODE_ID, 1124](#)
- [_NG_NODE_IS_VALID, 1124](#)
- [_NG_NODE_NAME, 1124](#)
- [_NG_NODE_NOT_VALID, 1124](#)
- [_NG_NODE_NUMHOOKS, 1124](#)
- [_NG_NODE_PRIVATE, 1124](#)
- [_NG_NODE_REALLY_DIE, 1124](#)
- [_NG_NODE_REF, 1125](#)
- [_NG_NODE_REVIVE, 1125](#)
- [_NG_NODE_SET_PRIVATE, 1125](#)
- [_NG_NODE_UNREF, 1125](#)
- [HK_DEAD, 1127](#)
- [HK_FORCE_WRITER, 1127](#)
- [HK_INVALID, 1127](#)
- [HK_QUEUE, 1127](#)
- [hook_p, 1150](#)
- [item_p, 1150](#)
- [MALLOC_DECLARE, 1151](#)
- [meta_p, 1150](#)
- [NETGRAPH_INIT, 1127](#)
- [NETGRAPH_INIT_ORDERED, 1127](#)
- [NG_ABI_VERSION, 1127](#)
- [ng_address_hook, 1151](#)
- [ng_address_ID, 1151](#)
- [ng_address_path, 1151](#)
- [ng_apply_t, 1150](#)
- [ng_bypass, 1151](#)
- [ng_callout, 1152](#)
- [ng_callout_init, 1128](#)
- [ng_close_t, 1150](#)
- [NG_CLOSING, 1128](#)
- [ng_connect_t, 1150](#)
- [ng_constructor_t, 1150](#)
- [ng_copy_meta, 1128](#)
- [ng_disconnect_t, 1150](#)
- [ng_findhook, 1152](#)
- [ng_findhook_t, 1150](#)
- [ng_findtype, 1152](#)
- [ng_fn_eachhook, 1150](#)
- [NG_FORCE_WRITER, 1128](#)
- [NG_FREE_ITEM, 1128](#)
- [ng_free_item, 1152](#)
- [NG_FREE_M, 1129](#)
- [NG_FREE_META, 1129](#)
- [NG_FREE_MSG, 1130](#)
- [NG_FWD_ITEM_HOOK, 1130](#)
- [NG_FWD_ITEM_HOOK_FLAGS, 1130](#)
- [NG_FWD_MSG_HOOK, 1131](#)
- [NG_FWD_NEW_DATA, 1131](#)
- [NG_FWD_NEW_DATA_FLAGS, 1131](#)
- [NG_HOOK_FORCE_QUEUE, 1131](#)
- [NG_HOOK_FORCE_WRITER, 1131](#)
- [NG_HOOK_IS_VALID, 1132](#)
- [NG_HOOK_NAME, 1132](#)
- [NG_HOOK_NODE, 1132](#)
- [NG_HOOK_NOT_VALID, 1133](#)
- [NG_HOOK_PEER, 1133](#)
- [NG_HOOK_PRIVATE, 1133](#)
- [NG_HOOK_REF, 1133](#)
- [NG_HOOK_SET_PRIVATE, 1134](#)
- [NG_HOOK_SET_RCVDATA, 1134](#)
- [NG_HOOK_SET_RCVMSG, 1134](#)
- [NG_HOOK_UNREF, 1134](#)
- [NG_INVALID, 1134](#)
- [ng_item_fn, 1150](#)
- [ng_make_node_common, 1152](#)
- [ng_mod_event, 1153](#)
- [ng_name_node, 1153](#)
- [ng_newhook_t, 1150](#)
- [ng_newtype, 1153](#)
- [ng_node2ID, 1154](#)
- [NG_NODE_FORCE_WRITER, 1134](#)
- [NG_NODE_FOREACH_HOOK, 1135](#)
- [NG_NODE_HAS_NAME, 1135](#)
- [NG_NODE_ID, 1135](#)
- [NG_NODE_IS_VALID, 1135](#)
- [NG_NODE_NAME, 1135](#)
- [NG_NODE_NOT_VALID, 1136](#)
- [NG_NODE_NUMHOOKS, 1136](#)
- [NG_NODE_PRIVATE, 1136](#)
- [NG_NODE_REALLY_DIE, 1138](#)
- [NG_NODE_REF, 1138](#)
- [NG_NODE_REVIVE, 1138](#)
- [NG_NODE_SET_PRIVATE, 1138](#)

- NG_NODE_UNREF, 1139
- NG_NOFLAGS, 1139
- ng_package_data, 1154
- ng_package_msg, 1154
- ng_package_msg_self, 1154
- NG_PEER_HOOK_NAME, 1139
- NG_PEER_NODE, 1140
- NG_PEER_NODE_NAME, 1140
- NG_PRIO_CUTOFF, 1140
- NG_PRIO_LINKSTATE, 1140
- NG_PROGRESS, 1140
- NG_QUEUE, 1140
- ng_rcvdata_t, 1151
- ng_rcvitem, 1151
- ng_rcvmsg_t, 1151
- NG REALLY_DIE, 1140
- ng_replace_retaddr, 1155
- NG_RESPOND_MSG, 1140
- ng_rmhook_self, 1155
- ng_rmnode_self, 1155
- ng_rmtime, 1156
- NG_SEND_DATA, 1141
- NG_SEND_DATA_FLAGS, 1141
- NG_SEND_DATA_ONLY, 1141
- ng_send_fn, 1142
- ng_send_fn1, 1156
- NG_SEND_MSG_HOOK, 1142
- NG_SEND_MSG_ID, 1142
- NG_SEND_MSG_PATH, 1143
- NG_SEPARATE_MALLOC, 1143
- ng_shutdown_t, 1151
- ng_snd_item, 1156
- NG_TAG_PRIO, 1143
- ng_uncallout, 1157
- ng_unref_hook, 1157
- ng_unref_node, 1157
- NG_WAITOK, 1143
- NG_WORKQ, 1144
- NGF_CLOSING, 1144
- NGF_FORCE_WRITER, 1144
- NGF_INVALID, 1144
- NGF REALLY_DIE, 1144
- NGF_TYPE1, 1144
- NGF_TYPE2, 1144
- NGF_TYPE3, 1144
- NGF_TYPE4, 1144
- NGF_WORKQ, 1144
- NGI_ARG1, 1145
- NGI_ARG2, 1145
- NGI_CLR_HOOK, 1145
- NGI_CLR_NODE, 1145
- NGI_FN, 1145
- NGI_GET_HOOK, 1145
- NGI_GET_M, 1145
- NGI_GET_META, 1146
- NGI_GET_MSG, 1146
- NGI_GET_NODE, 1146
- NGI_HOOK, 1147
- NGI_M, 1147
- NGI_META, 1147
- NGI_MSG, 1147
- NGI_NODE, 1147
- NGI_QUEUED_READER, 1147
- NGI_QUEUED_WRITER, 1147
- NGI_RETADDR, 1147
- NGI_SET_HOOK, 1148
- NGI_SET_NODE, 1148
- NGI_SET_READER, 1148
- NGI_SET_WRITER, 1148
- NGQF_DATA, 1148
- NGQF_FN, 1148
- NGQF_MESG, 1148
- NGQF_QMODE, 1148
- NGQF_QREADER, 1149
- NGQF_QWRITER, 1149
- NGQF_READER, 1149
- NGQF_RW, 1149
- NGQF_TYPE, 1149
- NGQF_UNDEF, 1149
- NGQF_WRITER, 1149
- node_p, 1151
- SAVE_LINE, 1149
- NETGRAPH_INIT
 - netgraph.h, 1127
 - ng_async.c, 1160
 - ng_atm.c, 571
 - ng_atmllc.c, 1170
 - ng_atmpif.c, 535
 - ng_bpf.c, 1206
 - ng_bridge.c, 1217
 - ng_ccatm.c, 555
 - ng_cisco.c, 1234
 - ng_deflate.c, 1242
 - ng_device.c, 1252
 - ng_echo.c, 1258
 - ng_eiface.c, 1263
 - ng_etf.c, 1272
 - ng_ether.c, 1281
 - ng_fec.c, 1291
 - ng_frame_relay.c, 1303
 - ng_gif.c, 1311
 - ng_gif_demux.c, 1320
 - ng_h4.c, 680
 - ng_hci_main.c, 743
 - ng_hole.c, 1327
 - ng_hub.c, 1333
 - ng_iface.c, 1339
 - ng_ip_input.c, 1348

- ng_ipfw.c, 1353
- ng_ksocket.c, 1363
- ng_l2cap_main.c, 961
- ng_l2tp.c, 1379
- ng_lmi.c, 1398
- ng_mppc.c, 1423
- ng_nat.c, 1432
- ng_netflow.c, 1103
- ng_one2many.c, 1438
- ng_ppp.c, 1493
- ng_pppoe.c, 1521
- ng_pptpgre.c, 1544
- ng_pred1.c, 1557
- ng_rfc1490.c, 1567
- ng_sample.c, 1575
- ng_source.c, 1599
- ng_split.c, 1608
- ng_sppp.c, 1615
- ng_sscfu.c, 613
- ng_sscop.c, 624
- ng_tag.c, 1623
- ng_tcpmss.c, 1632
- ng_tee.c, 1639
- ng_tty.c, 1648
- ng_UI.c, 1657
- ng_uni.c, 643
- ng_vjc.c, 1663
- ng_vlan.c, 1673
- NETGRAPH_INIT_ORDERED
 - netgraph.h, 1127
 - ng_socket.c, 1583
- netgraph_mod
 - ng_base.c, 1200
- NETISR_FEC
 - ng_fec.h, 1298
- netmask
 - ng_cisco_ipaddr, 189
- newhook
 - ng_type, 405
- next_hop
 - flow_entry_data, 78
 - netflow_v1_record, 108
 - netflow_v5_record, 114
- NEXT_QUEUED_ITEM_CAN_PROCEED
 - ng_base.c, 1179
- next_tag
 - ng_pppoe.c, 1522
- nextMany
 - ng_one2many_private, 345
- nfinfo_act_exp
 - ng_netflow_info, 333
- nfinfo_act_t
 - ng_netflow_info, 333
- nfinfo_alloc_failed
 - ng_netflow_info, 333
- nfinfo_bytes
 - ng_netflow_info, 333
- nfinfo_export_failed
 - ng_netflow_info, 333
- nfinfo_inact_exp
 - ng_netflow_info, 333
- nfinfo_inact_t
 - ng_netflow_info, 333
- nfinfo_packets
 - ng_netflow_info, 334
- nfinfo_used
 - ng_netflow_info, 334
- NG_ABI_VERSION
 - netgraph.h, 1127
- ng_acquire_read
 - ng_base.c, 1183
- ng_acquire_write
 - ng_base.c, 1183
- ng_add_hook
 - ng_base.c, 1184
- ng_address_hook
 - netgraph.h, 1151
 - ng_base.c, 1184
- ng_address_ID
 - netgraph.h, 1151
 - ng_base.c, 1184
- ng_address_path
 - netgraph.h, 1151
 - ng_base.c, 1184
- NG_ALLOC_HOOK
 - ng_base.c, 1180
- NG_ALLOC_NODE
 - ng_base.c, 1180
- ng_apply_item
 - ng_base.c, 1184
- ng_apply_t
 - netgraph.h, 1150
- ng_array_getAlign
 - ng_parse.c, 1452
- ng_array_getDefault
 - ng_parse.c, 1452
- ng_array_parse
 - ng_parse.c, 1453
- ng_array_unparse
 - ng_parse.c, 1453
- ng_async.c
 - ADD_BYTE, 1159
 - ASYNC_BUF_SIZE, 1159
 - ERROUT, 1159
 - fcstab, 1162
 - M_NETGRAPH_ASYNC, 1160
 - MODE_ESC, 1160
 - MODE_HUNT, 1160

- MODE_NORMAL, 1160
- NETGRAPH_INIT, 1160
- nga_async_add, 1160
- nga_cmdlist, 1162
- nga_config_type, 1162
- nga_config_type_fields, 1163
- nga_constructor, 1160, 1163
- nga_disconnect, 1161, 1163
- nga_newhook, 1161, 1163
- nga_rcv_async, 1161
- nga_rcv_sync, 1161
- nga_rcvdata, 1161, 1163
- nga_rcvmsg, 1162, 1163
- nga_shutdown, 1162, 1163
- nga_stats_type, 1163
- nga_stats_type_fields, 1163
- sc_p, 1160
- SYNC_BUF_SIZE, 1160
- typestruct, 1163
- ng_async.h
 - NGM_ASYNC_CMD_CLR_STATS, 1167
 - NGM_ASYNC_CMD_GET_CONFIG, 1167
 - NGM_ASYNC_CMD_GET_STATS, 1167
 - NGM_ASYNC_CMD_SET_CONFIG, 1167
- ng_async.h
 - NG_ASYNC_CONFIG_TYPE_INFO, 1165
 - NG_ASYNC_DEFAULT_MRU, 1165
 - NG_ASYNC_HOOK_ASYNC, 1165
 - NG_ASYNC_HOOK_SYNC, 1166
 - NG_ASYNC_MAX_MRU, 1166
 - NG_ASYNC_MIN_MRU, 1166
 - NG_ASYNC_NODE_TYPE, 1166
 - NG_ASYNC_STATS_TYPE_INFO, 1166
 - NGM_ASYNC_COOKIE, 1166
- ng_async_cfg, 116
 - accm, 116
 - amru, 116
 - enabled, 116
 - smru, 116
- NG_ASYNC_CONFIG_TYPE_INFO
 - ng_async.h, 1165
- NG_ASYNC_DEFAULT_MRU
 - ng_async.h, 1165
- NG_ASYNC_HOOK_ASYNC
 - ng_async.h, 1165
- NG_ASYNC_HOOK_SYNC
 - ng_async.h, 1166
- NG_ASYNC_MAX_MRU
 - ng_async.h, 1166
- NG_ASYNC_MIN_MRU
 - ng_async.h, 1166
- NG_ASYNC_NODE_TYPE
 - ng_async.h, 1166
- ng_async_private, 117
- abuf, 117
- amode, 118
- async, 118
- cfg, 118
- fcs, 118
- lasttime, 118
- node, 118
- sbuf, 118
- slen, 118
- stats, 118
- sync, 119
- ng_async_stat, 120
 - asyncBadCheckSums, 120
 - asyncFrames, 120
 - asyncOctets, 120
 - asyncOverflows, 120
 - asyncRunts, 120
 - syncFrames, 120
 - syncOctets, 121
 - syncOverflows, 121
- NG_ASYNC_STATS_TYPE_INFO
 - ng_async.h, 1166
- ng_atm.c
 - __FBSDID, 571
 - atmmmedia, 576
 - cpcs_term, 571
 - IFFLAGS, 570
 - IFP2NG, 570
 - IFP2NG_SET, 570
 - media, 576
 - name, 576
 - NETGRAPH_INIT, 571
 - ng_atm_acr_change_info, 576
 - ng_atm_acr_change_type, 576
 - ng_atm_attach, 571
 - ng_atm_attach_p, 576
 - ng_atm_cmdlist, 576
 - ng_atm_config_type, 576
 - ng_atm_config_type_info, 577
 - ng_atm_connect, 571, 577
 - ng_atm_constructor, 571, 577
 - ng_atm_cpcs_init, 572
 - ng_atm_cpcs_init_type, 577
 - ng_atm_cpcs_init_type_info, 577
 - ng_atm_cpcs_term, 572
 - ng_atm_cpcs_term_type, 577
 - ng_atm_cpcs_term_type_info, 577
 - ng_atm_detach, 572
 - ng_atm_detach_p, 578
 - ng_atm_disconnect, 572, 578
 - ng_atm_event, 573
 - ng_atm_event_func, 573
 - ng_atm_event_p, 578
 - ng_atm_if_change_info, 578

- ng_atm_if_change_type, 578
- ng_atm_input, 573
- ng_atm_input_orphan_p, 578
- ng_atm_input_orphans, 573
- ng_atm_input_p, 578
- ng_atm_mod_event, 574
- ng_atm_newhook, 574, 578
- ng_atm_output, 574
- ng_atm_output_p, 579
- ng_atm_rcvdata, 574, 579
- ng_atm_rcvdrop, 575, 579
- ng_atm_rcvmsg, 575, 579
- ng_atm_shutdown, 575, 579
- ng_atm_stats_type, 579
- ng_atm_stats_type_info, 579
- ng_atm_tparam_type, 579
- ng_atm_tparam_type_info, 580
- ng_atm_tpestruct, 580
- ng_atm_vcc_change_info, 580
- ng_atm_vcc_change_type, 580
- ng_atm_vcc_type, 581
- ng_atm_vcc_type_info, 581
- ng_atm_vccarray_getlen, 575
- ng_atm_vccarray_info, 581
- ng_atm_vccarray_type, 581
- ng_atm_vcctable_type, 581
- ng_atm_vcctable_type_info, 581
- SYSCTL_NODE, 575
- text_status, 575
- VCC_OPEN, 571
- ng_atm.h
 - NGM_ATM_ACR_CHANGE, 587
 - NGM_ATM_CARRIER_CHANGE, 587
 - NGM_ATM_CPCS_INIT, 587
 - NGM_ATM_CPCS_TERM, 587
 - NGM_ATM_GET_CONFIG, 587
 - NGM_ATM_GET_IFNAME, 587
 - NGM_ATM_GET_STATS, 587
 - NGM_ATM_GET_VCC, 587
 - NGM_ATM_GET_VCCID, 587
 - NGM_ATM_GET_VCCS, 587
 - NGM_ATM_IF_CHANGE, 587
 - NGM_ATM_VCC_CHANGE, 587
- ng_atm.h
 - NG_ATM_NODE_TYPE, 583
 - NGM_ATM_ACR_CHANGE_INFO, 583
 - NGM_ATM_CONFIG_INFO, 584
 - NGM_ATM_COOKIE, 584
 - NGM_ATM_CPCS_INIT_INFO, 584
 - NGM_ATM_CPCS_TERM_INFO, 585
 - NGM_ATM_IF_CHANGE_INFO, 585
 - NGM_ATM_STATS_INFO, 585
 - NGM_ATM_TPARAM_INFO, 585
 - NGM_ATM_VCC_CHANGE_INFO, 586
 - NGM_ATM_VCC_INFO, 586
 - NGM_ATM_VCCARRAY_INFO, 586
 - NGM_ATM_VCCTABLE_INFO, 586
- ng_atm_acr_change_info
 - ng_atm.c, 576
- ng_atm_acr_change_type
 - ng_atm.c, 576
- ng_atm_attach
 - ng_atm.c, 571
- ng_atm_attach_p
 - ng_atm.c, 576
- ng_atm_cmdlist
 - ng_atm.c, 576
- ng_atm_config_type
 - ng_atm.c, 576
- ng_atm_config_type_info
 - ng_atm.c, 577
- ng_atm_connect
 - ng_atm.c, 571, 577
- ng_atm_constructor
 - ng_atm.c, 571, 577
- ng_atm_cpcs_init
 - ng_atm.c, 572
- ng_atm_cpcs_init_type
 - ng_atm.c, 577
- ng_atm_cpcs_init_type_info
 - ng_atm.c, 577
- ng_atm_cpcs_term
 - ng_atm.c, 572
- ng_atm_cpcs_term_type
 - ng_atm.c, 577
- ng_atm_cpcs_term_type_info
 - ng_atm.c, 577
- ng_atm_detach
 - ng_atm.c, 572
- ng_atm_detach_p
 - ng_atm.c, 578
- ng_atm_disconnect
 - ng_atm.c, 572, 578
- ng_atm_event
 - ng_atm.c, 573
- ng_atm_event_func
 - ng_atm.c, 573
- ng_atm_event_p
 - ng_atm.c, 578
- ng_atm_if_change_info
 - ng_atm.c, 578
- ng_atm_if_change_type
 - ng_atm.c, 578
- ng_atm_input
 - ng_atm.c, 573
- ng_atm_input_orphan_p
 - ng_atm.c, 578
- ng_atm_input_orphans

- ng_atm.c, 573
- ng_atm_input_p
 - ng_atm.c, 578
- ng_atm_mod_event
 - ng_atm.c, 574
- ng_atm_newhook
 - ng_atm.c, 574, 578
- NG_ATM_NODE_TYPE
 - ng_atm.h, 583
- ng_atm_output
 - ng_atm.c, 574
- ng_atm_output_p
 - ng_atm.c, 579
- ng_atm_rcvdata
 - ng_atm.c, 574, 579
- ng_atm_rcvdrop
 - ng_atm.c, 575, 579
- ng_atm_rcvmsg
 - ng_atm.c, 575, 579
- ng_atm_shutdown
 - ng_atm.c, 575, 579
- ng_atm_stats_type
 - ng_atm.c, 579
- ng_atm_stats_type_info
 - ng_atm.c, 579
- ng_atm_tparam_type
 - ng_atm.c, 579
- ng_atm_tparam_type_info
 - ng_atm.c, 580
- ng_atm_typestruct
 - ng_atm.c, 580
- ng_atm_vcc_change_info
 - ng_atm.c, 580
- ng_atm_vcc_change_type
 - ng_atm.c, 580
- ng_atm_vcc_type
 - ng_atm.c, 581
- ng_atm_vcc_type_info
 - ng_atm.c, 581
- ng_atm_vccarray_getlen
 - ng_atm.c, 575
- ng_atm_vccarray_info
 - ng_atm.c, 581
- ng_atm_vccarray_type
 - ng_atm.c, 581
- ng_atm_vcctable_type
 - ng_atm.c, 581
- ng_atm_vcctable_type_info
 - ng_atm.c, 581
- ng_atmllc.c
 - NETGRAPH_INIT, 1170
 - ng_atmllc_constructor, 1170, 1171
 - ng_atmllc_disconnect, 1170, 1171
 - NG_ATMLLC_HEADER, 1169
 - NG_ATMLLC_HEADER_LEN, 1169
 - ng_atmllc_newhook, 1170, 1171
 - ng_atmllc_rcvdata, 1170, 1171
 - ng_atmllc_rcvmsg, 1171
 - ng_atmllc_shutdown, 1171
 - NG_ATMLLC_TYPE_ETHERNET_FCS, 1169
 - NG_ATMLLC_TYPE_ETHERNET_NOFCS, 1169
 - NG_ATMLLC_TYPE_FDDI_FCS, 1170
 - NG_ATMLLC_TYPE_FDDI_NOFCS, 1170
 - ng_atmllc_typestruct, 1171
- ng_atmllc.h
 - NG_ATMLLC_HOOK_802_4, 1173
 - NG_ATMLLC_HOOK_802_5, 1173
 - NG_ATMLLC_HOOK_802_6, 1173
 - NG_ATMLLC_HOOK_ATM, 1173
 - NG_ATMLLC_HOOK_BPDU, 1173
 - NG_ATMLLC_HOOK_ETHER, 1173
 - NG_ATMLLC_HOOK_FDDI, 1173
 - NG_ATMLLC_NODE_TYPE, 1174
 - NGM_ATMLLC_COOKIE, 1174
- ng_atmllc_constructor
 - ng_atmllc.c, 1170, 1171
- ng_atmllc_disconnect
 - ng_atmllc.c, 1170, 1171
- NG_ATMLLC_HEADER
 - ng_atmllc.c, 1169
- NG_ATMLLC_HEADER_LEN
 - ng_atmllc.c, 1169
- NG_ATMLLC_HOOK_802_4
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_802_5
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_802_6
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_ATM
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_BPDU
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_ETHER
 - ng_atmllc.h, 1173
- NG_ATMLLC_HOOK_FDDI
 - ng_atmllc.h, 1173
- ng_atmllc_newhook
 - ng_atmllc.c, 1170, 1171
- NG_ATMLLC_NODE_TYPE
 - ng_atmllc.h, 1174
- ng_atmllc_priv, 122
 - atm, 122
 - ether, 122
 - fddi, 122
- ng_atmllc_rcvdata
 - ng_atmllc.c, 1170, 1171

- ng_atmllc_rcvmsg
 - ng_atmllc.c, 1171
- ng_atmllc_shutdown
 - ng_atmllc.c, 1171
- NG_ATMLLC_TYPE_ETHERNET_FCS
 - ng_atmllc.c, 1169
- NG_ATMLLC_TYPE_ETHERNET_NOFCS
 - ng_atmllc.c, 1169
- NG_ATMLLC_TYPE_FDDI_FCS
 - ng_atmllc.c, 1170
- NG_ATMLLC_TYPE_FDDI_NOFCS
 - ng_atmllc.c, 1170
- ng_atmllc_tpestruct
 - ng_atmllc.c, 1171
- ng_atmpif.c
 - __FBSDID, 535
 - M_NETGRAPH_ATMPIF, 535
 - NETGRAPH_INIT, 535
 - ng_atmpif_cmdlist, 537
 - ng_atmpif_config_type, 537
 - ng_atmpif_config_type_fields, 537
 - ng_atmpif_constructor, 535, 537
 - ng_atmpif_disconnect, 535, 537
 - ng_atmpif_link_status_type, 537
 - ng_atmpif_link_status_type_fields, 537
 - ng_atmpif_mod_event, 535
 - ng_atmpif_newhook, 535, 538
 - ng_atmpif_rcvdata, 536, 538
 - ng_atmpif_rcvmsg, 536, 538
 - ng_atmpif_rmnode, 536, 538
 - ng_atmpif_stats_type, 538
 - ng_atmpif_stats_type_fields, 538
 - ng_atmpif_transmit, 536
 - ng_atmpif_tpestruct, 538
 - ng_mac_addr_type, 539
 - ng_macaddr_parse, 536, 539
 - ng_macaddr_unparse, 537, 539
 - vatmpif_nif_zone, 539
 - vatmpif_vcc_zone, 539
- ng_atmpif.h
 - NGM_ATMPIF_CLR_STATS, 592
 - NGM_ATMPIF_GET_CONFIG, 592
 - NGM_ATMPIF_GET_LINK_STATUS, 592
 - NGM_ATMPIF_GET_STATS, 592
 - NGM_ATMPIF_GETCLR_STATS, 592
 - NGM_ATMPIF_SET_CONFIG, 592
- ng_atmpif.h
 - Hva_Stats_aal5, 591
 - HVA_STATS_AAL5_NG_TYPE_INFO, 589
 - Hva_Stats_atm, 591
 - HVA_STATS_ATM_NG_TYPE_INFO, 589
 - Hva_Stats_ng, 591
 - HVA_STATS_NG_TYPE_INFO, 589
 - IS_VATMPIF_DEBUG_PACKET, 589
 - NG_ATMPIF_CONFIG_TYPE_INFO, 589
 - NG_ATMPIF_HOOK_LINK, 590
 - NG_ATMPIF_LINK_STATUS_TYPE_INFO, 590
 - NG_ATMPIF_NODE_TYPE, 590
 - NG_ATMPIF_STATS_TYPE_INFO, 590
 - NGM_ATMPIF_COOKIE, 590
 - VATMPIF_DEBUG_NONE, 590
 - VATMPIF_DEBUG_PACKET, 591
 - VATMPIF_DEV_NAME, 591
 - VATMPIF_MAX_VCI, 591
 - VATMPIF_MAX_VPI, 591
 - Vatmpif_stats, 591
- ng_atmpif_cmdlist
 - ng_atmpif.c, 537
- ng_atmpif_config_type
 - ng_atmpif.c, 537
- ng_atmpif_config_type_fields
 - ng_atmpif.c, 537
- NG_ATMPIF_CONFIG_TYPE_INFO
 - ng_atmpif.h, 589
- ng_atmpif_constructor
 - ng_atmpif.c, 535, 537
- ng_atmpif_disconnect
 - ng_atmpif.c, 535, 537
- ng_atmpif_harp.c
 - __FBSDID, 542
 - vatmpif_bearerclass, 542
 - vatmpif_harp_attach, 542
 - vatmpif_harp_closevcc, 543
 - vatmpif_harp_detach, 543
 - vatmpif_harp_instvcc, 543
 - vatmpif_harp_ioctl, 543
 - vatmpif_harp_openvcc, 543
 - vatmpif_harp_output, 544
 - vatmpif_harp_recv_drain, 544
 - vatmpif_harp_recv_stack, 544
 - vatmpif_nunits, 544
 - vatmpif_services, 544
 - vatmpif_svaal0, 545
 - vatmpif_svaal4, 545
 - vatmpif_svaal5, 545
- NG_ATMPIF_HOOK_LINK
 - ng_atmpif.h, 590
- ng_atmpif_link_status, 124
 - cur_pcr, 124
 - InSeq, 124
 - OutSeq, 124
- ng_atmpif_link_status_type
 - ng_atmpif.c, 537
- ng_atmpif_link_status_type_fields
 - ng_atmpif.c, 537
- NG_ATMPIF_LINK_STATUS_TYPE_INFO
 - ng_atmpif.h, 590

- ng_atmpif_mod_event
 - ng_atmpif.c, 535
- ng_atmpif_newhook
 - ng_atmpif.c, 535, 538
- NG_ATMPIF_NODE_TYPE
 - ng_atmpif.h, 590
- ng_atmpif_rcvdata
 - ng_atmpif.c, 536, 538
- ng_atmpif_rcvmsg
 - ng_atmpif.c, 536, 538
- ng_atmpif_rmnode
 - ng_atmpif.c, 536, 538
- ng_atmpif_stats_type
 - ng_atmpif.c, 538
- ng_atmpif_stats_type_fields
 - ng_atmpif.c, 538
- NG_ATMPIF_STATS_TYPE_INFO
 - ng_atmpif.h, 590
- ng_atmpif_transmit
 - ng_atmpif.c, 536
 - ng_atmpif_var.h, 550
- ng_atmpif_typestruct
 - ng_atmpif.c, 538
- ng_atmpif_var.h
 - VATMPIF_AAL_0, 550
 - VATMPIF_AAL_4, 550
 - VATMPIF_AAL_5, 550
 - VATMPIF_TRAF_ABR, 550
 - VATMPIF_TRAF_CBR, 550
 - VATMPIF_TRAF_UBR, 550
 - VATMPIF_TRAF_VBR, 550
- ng_atmpif_var.h
 - ng_atmpif_transmit, 550
 - ng_vatmpif_private, 547
 - priv_p, 549
 - Vatmpif_aal, 549
 - vatmpif_aal, 550
 - vatmpif_harp_attach, 550
 - vatmpif_harp_detach, 551
 - vatmpif_harp_rcv_drain, 551
 - vatmpif_nif_zone, 551
 - Vatmpif_traffic, 549
 - Vatmpif_traffic_type, 549
 - vatmpif_traffic_type, 550
 - Vatmpif_unit, 549
 - Vatmpif_vcc, 549
 - vatmpif_vcc_zone, 551
 - vu_closevcc, 547
 - vu_config, 547
 - vu_cur_pcr, 547
 - vu_flags, 547
 - vu_instvcc, 547
 - vu_ioctl, 547
 - vu_mtu, 548
 - vu_nif_zone, 548
 - vu_open_vcc, 548
 - vu_openvcc, 548
 - vu_output, 548
 - vu_pif, 548
 - vu_softc, 548
 - vu_stats, 548
 - vu_unit, 548
 - vu_vcc, 548
 - vu_vcc_zone, 548
 - vv_connvc, 549
 - vv_next, 549
 - vv_state, 549
 - vv_toku, 549
 - vv_upper, 549
- ng_attach_cntl
 - ng_socket.c, 1583
- ng_attach_common
 - ng_socket.c, 1583
- ng_attach_data
 - ng_socket.c, 1584
- ng_badpdu
 - hva_stats_ng, 93
- ng_base.c
 - _NG_ALLOC_HOOK, 1179
 - _NG_ALLOC_NODE, 1179
 - CHECK_DATA_MBUF, 1179
 - DECLARE_MODULE, 1183
 - DEFINE_PARSE_STRUCT_TYPE, 1179, 1183
 - HEAD_IS_READER, 1179
 - HEAD_IS_WRITER, 1179
 - ITEM_DEBUG_CHECKS, 1179
 - LIST_HEAD, 1183
 - maxalloc, 1200
 - MODULE_VERSION, 1183
 - netgraph_mod, 1200
 - NEXT_QUEUED_ITEM_CAN_PROCEED, 1179
 - ng_acquire_read, 1183
 - ng_acquire_write, 1183
 - ng_add_hook, 1184
 - ng_address_hook, 1184
 - ng_address_ID, 1184
 - ng_address_path, 1184
 - NG_ALLOC_HOOK, 1180
 - NG_ALLOC_NODE, 1180
 - ng_apply_item, 1184
 - ng_bypass, 1185
 - ng_callout, 1185
 - ng_callout_trampoline, 1186
 - ng_con_nodes, 1186
 - ng_con_part2, 1186
 - ng_con_part3, 1187

- ng_deadhook, 1201
- ng_deadnode, 1201
- ng_decodeidname, 1187
- ng_dequeue, 1187
- ng_destroy_hook, 1188
- ng_findhook, 1188
- ng_findtype, 1188
- ng_flush_input_queue, 1188
- NG_FREE_HOOK, 1180
- ng_free_item, 1189
- NG_FREE_NODE, 1180
- ng_generic_cmds, 1202
- ng_generic_linkinfo_array_type, 1202
- ng_generic_linkinfo_array_type_info, 1202
- ng_generic_linkinfo_getLength, 1189
- ng_generic_msg, 1189
- ng_generic_nodeinfoarray_type, 1202
- ng_generic_typeinfoarray_type, 1202
- ng_getqblk, 1190
- ng_ID2noderef, 1190
- NG_ID_HASH_SIZE, 1180
- NG_IDHASH_FIND, 1180
- NG_IDHASH_FN, 1180
- ng_leave_read, 1190
- ng_leave_write, 1190
- ng_macro_test, 1191
- ng_make_node, 1191
- ng_make_node_common, 1191
- ng_mkpeer, 1191
- ng_mod_event, 1192
- ng_name2noderef, 1192
- ng_name_node, 1192
- ng_newtype, 1193
- ng_node2ID, 1193
- ng_nodeinfoarray_type_info, 1203
- ng_package_data, 1193
- ng_package_msg, 1193
- ng_package_msg_self, 1194
- ng_path2noderef, 1194
- ng_path_parse, 1194
- ng_queue_rw, 1195
- ng_qzone, 1203
- ng_replace_retaddr, 1195
- ng_rmhook_part2, 1195
- ng_rmhook_self, 1195
- ng_rmnode, 1195
- ng_rmnode_self, 1196
- ng_rmtype, 1196
- ng_send_fn1, 1197
- ng_setisr, 1197
- ng_snd_item, 1197
- ng_typeinfoarray_type_info, 1203
- ng_uncallout, 1198
- ng_unname, 1198
- ng_unref_hook, 1198
- ng_unref_node, 1198
- ng_worklist_remove, 1198
- ngb_mod_event, 1199
- ngintr, 1199
- NGQ_RMASK, 1180
- NGQ_WMASK, 1181
- NGQRW_R, 1181
- NGQRW_W, 1181
- OP_PENDING, 1181
- QUEUE_ACTIVE, 1181
- QUEUED_READER_CAN_PROCEED, 1181
- QUEUED_WRITER_CAN_PROCEED, 1181
- READER_INCREMENT, 1181
- READER_MASK, 1182
- SAFETY_BARRIER, 1182
- SET_RETADDR, 1182
- SYSCTL_INT, 1200
- SYSCTL_NODE, 1200
- TAILQ_HEAD, 1200
- TESTING, 1182
- TRAP_ERROR, 1182
- TUNABLE_INT, 1200
- WRITER_ACTIVE, 1182
- ng_bind
 - ng_socket.c, 1584
- ng_bluetooth.c
 - bluetooth_hci_command_timeout, 658
 - bluetooth_hci_command_timeout_value, 660
 - bluetooth_hci_connect_timeout, 658
 - bluetooth_hci_connect_timeout_value, 660
 - bluetooth_hci_max_neighbor_age, 658
 - bluetooth_hci_max_neighbor_age_value, 660
 - bluetooth_l2cap_ertx_timeout, 658
 - bluetooth_l2cap_ertx_timeout_value, 661
 - bluetooth_l2cap_rtx_timeout, 659
 - bluetooth_l2cap_rtx_timeout_value, 661
 - bluetooth_mod, 661
 - bluetooth_modevent, 659
 - bluetooth_set_hci_command_timeout_value, 659
 - bluetooth_set_hci_connect_timeout_value, 659
 - bluetooth_set_l2cap_ertx_timeout_value, 659
 - bluetooth_set_l2cap_rtx_timeout_value, 659
 - DECLARE_MODULE, 659
 - MODULE_VERSION, 660
 - SYSCTL_INT, 660
 - SYSCTL_NODE, 660
 - SYSCTL_PROC, 660
- ng_bluetooth.h
 - bluetooth_hci_command_timeout, 789
 - bluetooth_hci_connect_timeout, 789
 - bluetooth_hci_max_neighbor_age, 789

- bluetooth_l2cap_ertx_timeout, 790
- bluetooth_l2cap_rtx_timeout, 790
- NG_BLUETOOTH_VERSION, 784
- NG_BT_ITEMQ_DEQUEUE, 784
- NG_BT_ITEMQ_DESTROY, 785
- NG_BT_ITEMQ_DRAIN, 785
- NG_BT_ITEMQ_DROP, 785
- NG_BT_ITEMQ_ENQUEUE, 785
- NG_BT_ITEMQ_FIRST, 786
- NG_BT_ITEMQ_FULL, 786
- NG_BT_ITEMQ_INIT, 786
- NG_BT_ITEMQ_LEN, 786
- ng_bt_itemq_p, 789
- NG_BT_ITEMQ_PREPEND, 786
- ng_bt_itemq_t, 789
- NG_BT_MBUFQ_DEQUEUE, 786
- NG_BT_MBUFQ_DESTROY, 787
- NG_BT_MBUFQ_DRAIN, 787
- NG_BT_MBUFQ_DROP, 787
- NG_BT_MBUFQ_ENQUEUE, 787
- NG_BT_MBUFQ_FIRST, 788
- NG_BT_MBUFQ_FULL, 788
- NG_BT_MBUFQ_INIT, 788
- NG_BT_MBUFQ_LEN, 788
- ng_bt_mbufq_p, 789
- NG_BT_MBUFQ_PREPEND, 788
- ng_bt_mbufq_t, 789
- NG_BLUETOOTH_VERSION
- ng_bluetooth.h, 784
- ng_bpf.c
 - ERROUT, 1205
 - hinfo_p, 1206
 - M_NETGRAPH_BPF, 1205
 - NETGRAPH_INIT, 1206
 - ng_bpf_cmdlist, 1208
 - ng_bpf_constructor, 1206, 1208
 - ng_bpf_default_prog, 1208
 - ng_bpf_disconnect, 1206, 1208
 - ng_bpf_hookprog_type, 1208
 - ng_bpf_hookprog_type_fields, 1208
 - ng_bpf_hookprogary_getLength, 1206
 - ng_bpf_hookprogary_info, 1208
 - ng_bpf_hookprogary_type, 1209
 - ng_bpf_hookstat_type, 1209
 - ng_bpf_hookstat_type_fields, 1209
 - ng_bpf_insn_type, 1209
 - ng_bpf_insn_type_fields, 1209
 - ng_bpf_newhook, 1206, 1210
 - ng_bpf_rcvdata, 1207, 1210
 - ng_bpf_rcvmsg, 1207, 1210
 - ng_bpf_setprog, 1207
 - ng_bpf_shutdown, 1207, 1210
 - OFFSETOF, 1206
 - typestruct, 1210
- ng_bpf.h
 - NGM_BPF_CLR_STATS, 1212
 - NGM_BPF_GET_PROGRAM, 1212
 - NGM_BPF_GET_STATS, 1212
 - NGM_BPF_GETCLR_STATS, 1212
 - NGM_BPF_SET_PROGRAM, 1212
- ng_bpf.h
 - NG_BPF_HOOKPROG_SIZE, 1211
 - NG_BPF_HOOKPROG_TYPE_INFO, 1211
 - NG_BPF_HOOKSTAT_TYPE_INFO, 1211
 - NG_BPF_NODE_TYPE, 1212
 - NGM_BPF_COOKIE, 1212
- ng_bpf_cmdlist
 - ng_bpf.c, 1208
- ng_bpf_constructor
 - ng_bpf.c, 1206, 1208
- ng_bpf_default_prog
 - ng_bpf.c, 1208
- ng_bpf_disconnect
 - ng_bpf.c, 1206, 1208
- ng_bpf_hookinfo, 125
 - hook, 125
 - node, 125
 - prog, 125
 - stats, 126
- ng_bpf_hookprog, 127
 - bpf_prog, 127
 - bpf_prog_len, 127
 - ifMatch, 127
 - ifNotMatch, 127
 - thisHook, 127
- NG_BPF_HOOKPROG_SIZE
 - ng_bpf.h, 1211
- ng_bpf_hookprog_type
 - ng_bpf.c, 1208
- ng_bpf_hookprog_type_fields
 - ng_bpf.c, 1208
- NG_BPF_HOOKPROG_TYPE_INFO
 - ng_bpf.h, 1211
- ng_bpf_hookprogary_getLength
 - ng_bpf.c, 1206
- ng_bpf_hookprogary_info
 - ng_bpf.c, 1208
- ng_bpf_hookprogary_type
 - ng_bpf.c, 1209
- ng_bpf_hookstat, 128
 - recvFrames, 128
 - recvMatchFrames, 128
 - recvMatchOctets, 128
 - recvOctets, 128
 - xmitFrames, 128
 - xmitOctets, 128
- ng_bpf_hookstat_type
 - ng_bpf.c, 1209

- ng_bpf_hookstat_type_fields
 - ng_bpf.c, 1209
- NG_BPF_HOOKSTAT_TYPE_INFO
 - ng_bpf.h, 1211
- ng_bpf_insn_type
 - ng_bpf.c, 1209
- ng_bpf_insn_type_fields
 - ng_bpf.c, 1209
- ng_bpf_newhook
 - ng_bpf.c, 1206, 1210
- NG_BPF_NODE_TYPE
 - ng_bpf.h, 1212
- ng_bpf_rcvdata
 - ng_bpf.c, 1207, 1210
- ng_bpf_rcvmsg
 - ng_bpf.c, 1207, 1210
- ng_bpf_setprog
 - ng_bpf.c, 1207
- ng_bpf_shutdown
 - ng_bpf.c, 1207, 1210
- ng_bridge.c
 - DEFAULT_LOOP_TIMEOUT, 1216
 - DEFAULT_MAX_STALENESS, 1216
 - DEFAULT_MIN_STABLE_AGE, 1216
 - ETHER_EQUAL, 1216
 - HASH, 1216
 - LINK_NUM, 1216
 - M_NETGRAPH_BRIDGE, 1216
 - MAX_BUCKETS, 1217
 - MIN_BUCKETS, 1217
 - NETGRAPH_INIT, 1217
 - ng_bridge_bcast_addr, 1221
 - ng_bridge_cmdlist, 1221
 - ng_bridge_config_type, 1221
 - ng_bridge_config_type_fields, 1221
 - ng_bridge_constructor, 1217, 1221
 - ng_bridge_disconnect, 1217, 1221
 - ng_bridge_get, 1218
 - ng_bridge_getTableLength, 1218
 - ng_bridge_hary_type, 1221
 - ng_bridge_hary_type_info, 1221
 - ng_bridge_host_ary_type, 1222
 - ng_bridge_host_ary_type_fields, 1222
 - ng_bridge_host_type, 1222
 - ng_bridge_host_type_fields, 1222
 - ng_bridge_ipfwary_type, 1222
 - ng_bridge_ipfwary_type_info, 1222
 - ng_bridge_newhook, 1218, 1223
 - ng_bridge_nodename, 1218
 - ng_bridge_put, 1218
 - ng_bridge_rcvdata, 1219, 1223
 - ng_bridge_rcvmsg, 1219, 1223
 - ng_bridge_rehash, 1219
 - ng_bridge_remove_hosts, 1220
 - ng_bridge_shutdown, 1220, 1223
 - ng_bridge_stats_type, 1223
 - ng_bridge_stats_type_fields, 1223
 - ng_bridge_timeout, 1220
 - ng_bridge_tpestruct, 1223
 - priv_p, 1217
 - SLIST_HEAD, 1220
- ng_bridge.h
 - NGM_BRIDGE_CLR_STATS, 1227
 - NGM_BRIDGE_GET_CONFIG, 1227
 - NGM_BRIDGE_GET_STATS, 1227
 - NGM_BRIDGE_GET_TABLE, 1227
 - NGM_BRIDGE_GETCLR_STATS, 1227
 - NGM_BRIDGE_RESET, 1227
 - NGM_BRIDGE_SET_CONFIG, 1227
- ng_bridge.h
 - NG_BRIDGE_CONFIG_TYPE_INFO, 1225
 - NG_BRIDGE_HOOK_LINK_FMT, 1225
 - NG_BRIDGE_HOOK_LINK_PREFIX, 1226
 - NG_BRIDGE_HOST_ARY_TYPE_INFO, 1226
 - NG_BRIDGE_HOST_TYPE_INFO, 1226
 - NG_BRIDGE_MAX_LINKS, 1226
 - NG_BRIDGE_NODE_TYPE, 1226
 - NG_BRIDGE_STATS_TYPE_INFO, 1226
 - NGM_BRIDGE_COOKIE, 1227
- ng_bridge_bcast_addr
 - ng_bridge.c, 1221
- ng_bridge_cmdlist
 - ng_bridge.c, 1221
- ng_bridge_config, 130
 - debugLevel, 130
 - ipfw, 130
 - loopTimeout, 130
 - maxStaleness, 130
 - minStableAge, 130
- ng_bridge_config_type
 - ng_bridge.c, 1221
- ng_bridge_config_type_fields
 - ng_bridge.c, 1221
- NG_BRIDGE_CONFIG_TYPE_INFO
 - ng_bridge.h, 1225
- ng_bridge_constructor
 - ng_bridge.c, 1217, 1221
- ng_bridge_disconnect
 - ng_bridge.c, 1217, 1221
- ng_bridge_get
 - ng_bridge.c, 1218
- ng_bridge_getTableLength
 - ng_bridge.c, 1218
- ng_bridge_hary_type
 - ng_bridge.c, 1221
- ng_bridge_hary_type_info
 - ng_bridge.c, 1221

- ng_bridge_hent, [131](#)
 - host, [131](#)
- NG_BRIDGE_HOOK_LINK_FMT
 - ng_bridge.h, [1225](#)
- NG_BRIDGE_HOOK_LINK_PREFIX
 - ng_bridge.h, [1226](#)
- ng_bridge_host, [132](#)
 - addr, [132](#)
 - age, [132](#)
 - linkNum, [132](#)
 - staleness, [132](#)
- ng_bridge_host_ary, [133](#)
 - hosts, [133](#)
 - numHosts, [133](#)
- ng_bridge_host_ary_type
 - ng_bridge.c, [1222](#)
- ng_bridge_host_ary_type_fields
 - ng_bridge.c, [1222](#)
- NG_BRIDGE_HOST_ARY_TYPE_INFO
 - ng_bridge.h, [1226](#)
- ng_bridge_host_type
 - ng_bridge.c, [1222](#)
- ng_bridge_host_type_fields
 - ng_bridge.c, [1222](#)
- NG_BRIDGE_HOST_TYPE_INFO
 - ng_bridge.h, [1226](#)
- ng_bridge_ipfwary_type
 - ng_bridge.c, [1222](#)
- ng_bridge_ipfwary_type_info
 - ng_bridge.c, [1222](#)
- ng_bridge_link, [134](#)
 - hook, [134](#)
 - loopCount, [134](#)
 - stats, [134](#)
- ng_bridge_link_stats, [136](#)
 - loopDetects, [136](#)
 - loopDrops, [136](#)
 - memoryFailures, [136](#)
 - recvBroadcasts, [136](#)
 - recvInvalid, [136](#)
 - recvMulticasts, [137](#)
 - recvOctets, [137](#)
 - recvPackets, [137](#)
 - recvRunts, [137](#)
 - recvUnknown, [137](#)
 - xmitBroadcasts, [137](#)
 - xmitMulticasts, [137](#)
 - xmitOctets, [137](#)
 - xmitPackets, [137](#)
- NG_BRIDGE_MAX_LINKS
 - ng_bridge.h, [1226](#)
- ng_bridge_newhook
 - ng_bridge.c, [1218](#), [1223](#)
- NG_BRIDGE_NODE_TYPE
 - ng_bridge.h, [1226](#)
- ng_bridge_nodename
 - ng_bridge.c, [1218](#)
- ng_bridge_private, [139](#)
 - conf, [139](#)
 - hashMask, [139](#)
 - links, [140](#)
 - node, [140](#)
 - numBuckets, [140](#)
 - numHosts, [140](#)
 - numLinks, [140](#)
 - tab, [140](#)
 - timer, [140](#)
- ng_bridge_put
 - ng_bridge.c, [1218](#)
- ng_bridge_rcvdata
 - ng_bridge.c, [1219](#), [1223](#)
- ng_bridge_rcvmsg
 - ng_bridge.c, [1219](#), [1223](#)
- ng_bridge_rehash
 - ng_bridge.c, [1219](#)
- ng_bridge_remove_hosts
 - ng_bridge.c, [1220](#)
- ng_bridge_shutdown
 - ng_bridge.c, [1220](#), [1223](#)
- ng_bridge_stats_type
 - ng_bridge.c, [1223](#)
- ng_bridge_stats_type_fields
 - ng_bridge.c, [1223](#)
- NG_BRIDGE_STATS_TYPE_INFO
 - ng_bridge.h, [1226](#)
- ng_bridge_timeout
 - ng_bridge.c, [1220](#)
- ng_bridge_ttypestruct
 - ng_bridge.c, [1223](#)
- ng_bt3c.h
 - NG_BT3C_ALERT_LEVEL, [791](#)
 - NG_BT3C_ERR_LEVEL, [791](#)
 - NG_BT3C_HOOK, [792](#)
 - NG_BT3C_INFO_LEVEL, [792](#)
 - ng_bt3c_node_debug_ep, [794](#)
 - ng_bt3c_node_state_ep, [794](#)
 - NG_BT3C_NODE_TYPE, [792](#)
 - NG_BT3C_W4_PKT_DATA, [792](#)
 - NG_BT3C_W4_PKT_HDR, [792](#)
 - NG_BT3C_W4_PKT_IND, [792](#)
 - NG_BT3C_WARN_LEVEL, [792](#)
 - NGM_BT3C_COOKIE, [792](#)
 - NGM_BT3C_NODE_DOWNLOAD_-
FIRMWARE, [792](#)
 - NGM_BT3C_NODE_GET_DEBUG, [792](#)
 - NGM_BT3C_NODE_GET_QLEN, [793](#)
 - NGM_BT3C_NODE_GET_STAT, [793](#)
 - NGM_BT3C_NODE_GET_STATE, [793](#)

- NGM_BT3C_NODE_IN_QUEUE, 793
- NGM_BT3C_NODE_OUT_QUEUE, 793
- NGM_BT3C_NODE_RESET_STAT, 793
- NGM_BT3C_NODE_SET_DEBUG, 793
- NGM_BT3C_NODE_SET_QLEN, 793
- NG_BT3C_ALERT
 - ng_bt3c_var.h, 675
- NG_BT3C_ALERT_LEVEL
 - ng_bt3c.h, 791
- ng_bt3c_cmdlist
 - ng_bt3c_pccard.c, 671
- ng_bt3c_connect
 - ng_bt3c_pccard.c, 669, 671
- ng_bt3c_constructor
 - ng_bt3c_pccard.c, 669, 671
- ng_bt3c_disconnect
 - ng_bt3c_pccard.c, 669, 671
- NG_BT3C_ERR
 - ng_bt3c_var.h, 675
- NG_BT3C_ERR_LEVEL
 - ng_bt3c.h, 791
- ng_bt3c_firmware_block_ep, 141
 - block_address, 141
 - block_alignment, 141
 - block_size, 141
- NG_BT3C_HOOK
 - ng_bt3c.h, 792
- NG_BT3C_INFO
 - ng_bt3c_var.h, 676
- NG_BT3C_INFO_LEVEL
 - ng_bt3c.h, 792
- ng_bt3c_newhook
 - ng_bt3c_pccard.c, 669, 671
- ng_bt3c_node_debug_ep
 - ng_bt3c.h, 794
- ng_bt3c_node_qlen_ep, 142
 - qlen, 142
 - queue, 142
- ng_bt3c_node_qlen_type
 - ng_bt3c_pccard.c, 671
- ng_bt3c_node_qlen_type_fields
 - ng_bt3c_pccard.c, 672
- ng_bt3c_node_stat_ep, 143
 - bytes_recv, 143
 - bytes_sent, 143
 - ierrors, 143
 - oerrors, 143
 - pckts_recv, 143
 - pckts_sent, 143
- ng_bt3c_node_stat_type
 - ng_bt3c_pccard.c, 672
- ng_bt3c_node_stat_type_fields
 - ng_bt3c_pccard.c, 672
- ng_bt3c_node_state_ep
 - ng_bt3c.h, 794
- NG_BT3C_NODE_TYPE
 - ng_bt3c.h, 792
- ng_bt3c_pccard.c
 - bt3c_devclass, 671
 - bt3c_download_firmware, 666
 - bt3c_forward, 666
 - bt3c_intr, 667
 - bt3c_modevent, 667
 - bt3c_pccard_attach, 667
 - bt3c_pccard_detach, 668
 - bt3c_pccard_driver, 671
 - bt3c_pccard_methods, 671
 - bt3c_pccard_probe, 668
 - bt3c_read, 665
 - bt3c_read_control, 665
 - bt3c_read_data, 665
 - bt3c_receive, 668
 - bt3c_send, 668
 - bt3c_set_address, 665
 - bt3c_swi_intr, 668
 - bt3c_write, 665
 - bt3c_write_control, 666
 - bt3c_write_data, 666
 - DRIVER_MODULE, 669
 - MALLOC_DEFINE, 669
 - MODULE_DEPEND, 669
 - MODULE_VERSION, 669
 - ng_bt3c_cmdlist, 671
 - ng_bt3c_connect, 669, 671
 - ng_bt3c_constructor, 669, 671
 - ng_bt3c_disconnect, 669, 671
 - ng_bt3c_newhook, 669, 671
 - ng_bt3c_node_qlen_type, 671
 - ng_bt3c_node_qlen_type_fields, 672
 - ng_bt3c_node_stat_type, 672
 - ng_bt3c_node_stat_type_fields, 672
 - ng_bt3c_rcvdata, 669, 672
 - ng_bt3c_rcvmsg, 670, 672
 - ng_bt3c_shutdown, 670, 673
 - typestruct, 673
- ng_bt3c_rcvdata
 - ng_bt3c_pccard.c, 669, 672
- ng_bt3c_rcvmsg
 - ng_bt3c_pccard.c, 670, 672
- ng_bt3c_shutdown
 - ng_bt3c_pccard.c, 670, 673
- NG_BT3C_STAT_BYTES_RECV
 - ng_bt3c_var.h, 676
- NG_BT3C_STAT_BYTES_SENT
 - ng_bt3c_var.h, 676
- NG_BT3C_STAT_IERROR
 - ng_bt3c_var.h, 676
- NG_BT3C_STAT_OERROR

- ng_bt3c_var.h, 676
- NG_BT3C_STAT_PCKTS_RECV
 - ng_bt3c_var.h, 676
- NG_BT3C_STAT_PCKTS_SENT
 - ng_bt3c_var.h, 676
- NG_BT3C_STAT_RESET
 - ng_bt3c_var.h, 676
- ng_bt3c_var.h
 - BT3C_ADDR_H, 674
 - BT3C_ADDR_L, 674
 - BT3C_ANTENNA_OUT, 675
 - BT3C_CONTROL, 675
 - BT3C_DATA_H, 675
 - BT3C_DATA_L, 675
 - BT3C_DEFAULTQLEN, 675
 - BT3C_FIFO_SIZE, 675
 - bt3c_softc_p, 677
 - bt3c_softc_t, 677
 - BT3C_XMIT, 675
 - NG_BT3C_ALERT, 675
 - NG_BT3C_ERR, 675
 - NG_BT3C_INFO, 676
 - NG_BT3C_STAT_BYTES_RECV, 676
 - NG_BT3C_STAT_BYTES_SENT, 676
 - NG_BT3C_STAT_IERROR, 676
 - NG_BT3C_STAT_OERROR, 676
 - NG_BT3C_STAT_PCKTS_RECV, 676
 - NG_BT3C_STAT_PCKTS_SENT, 676
 - NG_BT3C_STAT_RESET, 676
 - NG_BT3C_WARN, 676
- NG_BT3C_W4_PKT_DATA
 - ng_bt3c.h, 792
- NG_BT3C_W4_PKT_HDR
 - ng_bt3c.h, 792
- NG_BT3C_W4_PKT_IND
 - ng_bt3c.h, 792
- NG_BT3C_WARN
 - ng_bt3c_var.h, 676
- NG_BT3C_WARN_LEVEL
 - ng_bt3c.h, 792
- ng_bt_itemq, 144
 - drops, 144
 - head, 144
 - len, 144
 - maxlen, 145
 - tail, 145
- NG_BT_ITEMQ_DEQUEUE
 - ng_bluetooth.h, 784
- NG_BT_ITEMQ_DESTROY
 - ng_bluetooth.h, 785
- NG_BT_ITEMQ_DRAIN
 - ng_bluetooth.h, 785
- NG_BT_ITEMQ_DROP
 - ng_bluetooth.h, 785
- NG_BT_ITEMQ_ENQUEUE
 - ng_bluetooth.h, 785
- NG_BT_ITEMQ_FIRST
 - ng_bluetooth.h, 786
- NG_BT_ITEMQ_FULL
 - ng_bluetooth.h, 786
- NG_BT_ITEMQ_INIT
 - ng_bluetooth.h, 786
- NG_BT_ITEMQ_LEN
 - ng_bluetooth.h, 786
- ng_bt_itemq_p
 - ng_bluetooth.h, 789
- NG_BT_ITEMQ_PREPEND
 - ng_bluetooth.h, 786
- ng_bt_itemq_t
 - ng_bluetooth.h, 789
- ng_bt_mbufq, 146
 - drops, 146
 - head, 146
 - len, 146
 - maxlen, 146
 - tail, 146
- NG_BT_MBUFQ_DEQUEUE
 - ng_bluetooth.h, 786
- NG_BT_MBUFQ_DESTROY
 - ng_bluetooth.h, 787
- NG_BT_MBUFQ_DRAIN
 - ng_bluetooth.h, 787
- NG_BT_MBUFQ_DROP
 - ng_bluetooth.h, 787
- NG_BT_MBUFQ_ENQUEUE
 - ng_bluetooth.h, 787
- NG_BT_MBUFQ_FIRST
 - ng_bluetooth.h, 788
- NG_BT_MBUFQ_FULL
 - ng_bluetooth.h, 788
- NG_BT_MBUFQ_INIT
 - ng_bluetooth.h, 788
- NG_BT_MBUFQ_LEN
 - ng_bluetooth.h, 788
- ng_bt_mbufq_p
 - ng_bluetooth.h, 789
- NG_BT_MBUFQ_PREPEND
 - ng_bluetooth.h, 788
- ng_bt_mbufq_t
 - ng_bluetooth.h, 789
- ng_btsocket.c
 - DECLARE_MODULE, 1012
 - MODULE_DEPEND, 1012
 - MODULE_VERSION, 1012
 - ng_btsocket_domain, 1013
 - ng_btsocket_hci_raw_usrreqs, 1013
 - ng_btsocket_l2cap_raw_usrreqs, 1013
 - ng_btsocket_l2cap_usrreqs, 1014

- ng_btsocket_mod, 1014
- ng_btsocket_modevent, 1012
- ng_btsocket_protosw, 1014
- ng_btsocket_protosw_end, 1012
- ng_btsocket_protosw_size, 1012
- ng_btsocket_rfcomm_usrreqs, 1014
- SYSTL_NODE, 1012, 1013
- ng_btsocket.h
 - BLUETOOTH_PROTO_HCI, 797
 - BLUETOOTH_PROTO_L2CAP, 797
 - BLUETOOTH_PROTO_RFCOMM, 797
 - NG_BT_SOCKET_ALERT_LEVEL, 797
 - NG_BT_SOCKET_ERR_LEVEL, 797
 - NG_BT_SOCKET_HCI_RAW_NODE_TYPE, 797
 - NG_BT_SOCKET_INFO_LEVEL, 797
 - NG_BT_SOCKET_L2CAP_NODE_TYPE, 797
 - NG_BT_SOCKET_L2CAP_RAW_NODE_TYPE, 797
 - NG_BT_SOCKET_WARN_LEVEL, 797
 - SCM_HCI_RAW_DIRECTION, 798
 - SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_CACHE, 798
 - SIOC_HCI_RAW_NODE_GET_BDADDR, 798
 - SIOC_HCI_RAW_NODE_GET_BUFFER, 798
 - SIOC_HCI_RAW_NODE_GET_CON_LIST, 798
 - SIOC_HCI_RAW_NODE_GET_DEBUG, 798
 - SIOC_HCI_RAW_NODE_GET_FEATURES, 799
 - SIOC_HCI_RAW_NODE_GET_LINK_POLICY_MASK, 799
 - SIOC_HCI_RAW_NODE_GET_NEIGHBOR_CACHE, 799
 - SIOC_HCI_RAW_NODE_GET_PACKET_MASK, 799
 - SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH, 799
 - SIOC_HCI_RAW_NODE_GET_STAT, 800
 - SIOC_HCI_RAW_NODE_GET_STATE, 800
 - SIOC_HCI_RAW_NODE_INIT, 800
 - SIOC_HCI_RAW_NODE_LIST_NAMES, 800
 - SIOC_HCI_RAW_NODE_RESET_STAT, 800
 - SIOC_HCI_RAW_NODE_SET_DEBUG, 800
 - SIOC_HCI_RAW_NODE_SET_LINK_POLICY_MASK, 801
 - SIOC_HCI_RAW_NODE_SET_PACKET_MASK, 801
 - SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH, 801
 - SIOC_L2CAP_L2CA_GET_INFO, 801
 - SIOC_L2CAP_L2CA_PING, 801
 - SIOC_L2CAP_NODE_GET_AUTO_DISCON_TIMO, 802
 - SIOC_L2CAP_NODE_GET_CHAN_LIST, 802
 - SIOC_L2CAP_NODE_GET_CON_LIST, 802
 - SIOC_L2CAP_NODE_GET_DEBUG, 802
 - SIOC_L2CAP_NODE_GET_FLAGS, 802
 - SIOC_L2CAP_NODE_SET_AUTO_DISCON_TIMO, 803
 - SIOC_L2CAP_NODE_SET_DEBUG, 803
 - SO_HCI_RAW_DIRECTION, 803
 - SO_HCI_RAW_FILTER, 803
 - SO_L2CAP_FLUSH, 803
 - SO_L2CAP_IFLOW, 803
 - SO_L2CAP_IMTU, 804
 - SO_L2CAP_OFLOW, 804
 - SO_L2CAP_OMTU, 804
 - SO_RFCOMM_FC_INFO, 804
 - SO_RFCOMM_MTU, 804
 - SOL_HCI_RAW, 804
 - SOL_L2CAP, 804
 - SOL_RFCOMM, 804
 - NG_BT_SOCKET_ALERT_LEVEL
 - ng_btsocket.h, 797
 - ng_btsocket_domain
 - ng_btsocket.c, 1013
 - NG_BT_SOCKET_ERR_LEVEL
 - ng_btsocket.h, 797
 - ng_btsocket_hci_raw.c
 - ifqmaxlen, 1026
 - LIST_HEAD, 1020
 - M_NETGRAPH_BT_SOCKET_HCI_RAW, 1019
 - ng_btsocket_hci_raw_abort, 1020
 - NG_BT_SOCKET_HCI_RAW_ALERT, 1019
 - ng_btsocket_hci_raw_attach, 1020
 - ng_btsocket_hci_raw_bind, 1020
 - ng_btsocket_hci_raw_close, 1020
 - ng_btsocket_hci_raw_connect, 1020
 - ng_btsocket_hci_raw_control, 1021
 - ng_btsocket_hci_raw_ctloutput, 1021
 - ng_btsocket_hci_raw_data_input, 1022
 - ng_btsocket_hci_raw_debug_level, 1026
 - ng_btsocket_hci_raw_detach, 1022
 - ng_btsocket_hci_raw_disconnect, 1022
 - NG_BT_SOCKET_HCI_RAW_ERR, 1019
 - ng_btsocket_hci_raw_filter, 1022
 - ng_btsocket_hci_raw_get_token, 1022
 - NG_BT_SOCKET_HCI_RAW_INFO, 1019
 - ng_btsocket_hci_raw_init, 1022

- ng_btsocket_hci_raw_input, 1023
- ng_btsocket_hci_raw_ioctl_timeout, 1027
- ng_btsocket_hci_raw_msg_input, 1024
- ng_btsocket_hci_raw_node, 1027
- ng_btsocket_hci_raw_node_connect, 1024, 1027
- ng_btsocket_hci_raw_node_constructor, 1027
- ng_btsocket_hci_raw_node_disconnect, 1024, 1027
- ng_btsocket_hci_raw_node_newhook, 1024, 1027
- ng_btsocket_hci_raw_node_rcvdata, 1024, 1027
- ng_btsocket_hci_raw_node_rcvmsg, 1024, 1027
- ng_btsocket_hci_raw_node_shutdown, 1024, 1027
- ng_btsocket_hci_raw_output, 1025
- ng_btsocket_hci_raw_peeraddr, 1025
- ng_btsocket_hci_raw_queue, 1027
- ng_btsocket_hci_raw_queue_mtx, 1028
- ng_btsocket_hci_raw_savctl, 1025
- ng_btsocket_hci_raw_send, 1025
- ng_btsocket_hci_raw_send_ngmsg, 1026
- ng_btsocket_hci_raw_send_sync_ngmsg, 1026
- ng_btsocket_hci_raw_sockaddr, 1026
- ng_btsocket_hci_raw_task, 1028
- ng_btsocket_hci_raw_wakeup_input_task, 1019
- NG_BT_SOCKET_HCI_RAW_WARN, 1020
- typestruct, 1028
- ng_btsocket_hci_raw.h
 - ng_btsocket_hci_raw_abort, 806
 - ng_btsocket_hci_raw_attach, 806
 - ng_btsocket_hci_raw_bind, 807
 - ng_btsocket_hci_raw_close, 807
 - ng_btsocket_hci_raw_connect, 807
 - ng_btsocket_hci_raw_control, 807
 - ng_btsocket_hci_raw_ctloutput, 808
 - ng_btsocket_hci_raw_detach, 808
 - NG_BT_SOCKET_HCI_RAW_DIRECTION, 806
 - ng_btsocket_hci_raw_disconnect, 808
 - ng_btsocket_hci_raw_init, 808
 - ng_btsocket_hci_raw_pcb_p, 806
 - ng_btsocket_hci_raw_pcb_t, 806
 - ng_btsocket_hci_raw_peeraddr, 809
 - NG_BT_SOCKET_HCI_RAW_PRIVILEGED, 806
 - NG_BT_SOCKET_HCI_RAW_RECVSPACE, 806
 - ng_btsocket_hci_raw_send, 809
 - NG_BT_SOCKET_HCI_RAW_SENDSPEACE, 806
 - ng_btsocket_hci_raw_sockaddr, 810
 - so2hci_raw_pcb, 806
 - ng_btsocket_hci_raw_abort
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_hci_raw.h, 806
 - NG_BT_SOCKET_HCI_RAW_ALERT
 - ng_btsocket_hci_raw.c, 1019
 - ng_btsocket_hci_raw_attach
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_hci_raw.h, 806
 - ng_btsocket_hci_raw_bind
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_hci_raw.h, 807
 - ng_btsocket_hci_raw_close
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_hci_raw.h, 807
 - ng_btsocket_hci_raw_con_list, 147
 - connections, 147
 - num_connections, 147
 - ng_btsocket_hci_raw_connect
 - ng_btsocket_hci_raw.c, 1020
 - ng_btsocket_hci_raw.h, 807
 - ng_btsocket_hci_raw_control
 - ng_btsocket_hci_raw.c, 1021
 - ng_btsocket_hci_raw.h, 807
 - ng_btsocket_hci_raw_ctloutput
 - ng_btsocket_hci_raw.c, 1021
 - ng_btsocket_hci_raw.h, 808
 - ng_btsocket_hci_raw_data_input
 - ng_btsocket_hci_raw.c, 1022
 - ng_btsocket_hci_raw_debug_level
 - ng_btsocket_hci_raw.c, 1026
 - ng_btsocket_hci_raw_detach
 - ng_btsocket_hci_raw.c, 1022
 - ng_btsocket_hci_raw.h, 808
 - NG_BT_SOCKET_HCI_RAW_DIRECTION
 - ng_btsocket_hci_raw.h, 806
 - ng_btsocket_hci_raw_disconnect
 - ng_btsocket_hci_raw.c, 1022
 - ng_btsocket_hci_raw.h, 808
 - NG_BT_SOCKET_HCI_RAW_ERR
 - ng_btsocket_hci_raw.c, 1019
 - ng_btsocket_hci_raw_filter, 148
 - bit_decl, 148
 - ng_btsocket_hci_raw.c, 1022
 - ng_btsocket_hci_raw_get_token
 - ng_btsocket_hci_raw.c, 1022
 - NG_BT_SOCKET_HCI_RAW_INFO
 - ng_btsocket_hci_raw.c, 1019
 - ng_btsocket_hci_raw_init
 - ng_btsocket_hci_raw.c, 1022
 - ng_btsocket_hci_raw.h, 808

- ng_btsocket_hci_raw_input
 - ng_btsocket_hci_raw.c, 1023
- ng_btsocket_hci_raw_ioctl_timeout
 - ng_btsocket_hci_raw.c, 1027
- ng_btsocket_hci_raw_msg_input
 - ng_btsocket_hci_raw.c, 1024
- ng_btsocket_hci_raw_node
 - ng_btsocket_hci_raw.c, 1027
- ng_btsocket_hci_raw_node_bdaddr, 149
 - bdaddr, 149
- ng_btsocket_hci_raw_node_buffer, 150
 - buffer, 150
- ng_btsocket_hci_raw_node_connect
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_constructor
 - ng_btsocket_hci_raw.c, 1027
- ng_btsocket_hci_raw_node_debug, 151
 - debug, 151
- ng_btsocket_hci_raw_node_disconnect
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_features, 152
 - features, 152
- ng_btsocket_hci_raw_node_link_policy_mask, 153
 - policy_mask, 153
- ng_btsocket_hci_raw_node_list_names, 154
 - names, 154
 - num_names, 154
- ng_btsocket_hci_raw_node_neighbor_cache, 155
 - entries, 155
 - num_entries, 155
- ng_btsocket_hci_raw_node_newhook
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_packet_mask, 156
 - packet_mask, 156
- ng_btsocket_hci_raw_node_rcvdata
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_rcvmsg
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_role_switch, 157
 - role_switch, 157
- ng_btsocket_hci_raw_node_shutdown
 - ng_btsocket_hci_raw.c, 1024, 1027
- ng_btsocket_hci_raw_node_stat, 158
 - stat, 158
- ng_btsocket_hci_raw_node_state, 159
 - state, 159
- NG_BT_SOCKET_HCI_RAW_NODE_TYPE
 - ng_btsocket.h, 797
- ng_btsocket_hci_raw_output
 - ng_btsocket_hci_raw.c, 1025
- ng_btsocket_hci_raw_pcb, 160
 - addr, 160
 - filter, 160
 - flags, 161
 - LIST_ENTRY, 160
 - msg, 161
 - pcb_mtx, 161
 - so, 161
 - token, 161
- ng_btsocket_hci_raw_pcb_p
 - ng_btsocket_hci_raw.h, 806
- ng_btsocket_hci_raw_pcb_t
 - ng_btsocket_hci_raw.h, 806
- ng_btsocket_hci_raw_peeraddr
 - ng_btsocket_hci_raw.c, 1025
 - ng_btsocket_hci_raw.h, 809
- NG_BT_SOCKET_HCI_RAW_PRIVILEGED
 - ng_btsocket_hci_raw.h, 806
- ng_btsocket_hci_raw_queue
 - ng_btsocket_hci_raw.c, 1027
- ng_btsocket_hci_raw_queue_mtx
 - ng_btsocket_hci_raw.c, 1028
- NG_BT_SOCKET_HCI_RAW_RECVSPACE
 - ng_btsocket_hci_raw.h, 806
- ng_btsocket_hci_raw_savctl
 - ng_btsocket_hci_raw.c, 1025
- ng_btsocket_hci_raw_sec_filter, 162
 - bit_decl, 162
- ng_btsocket_hci_raw_send
 - ng_btsocket_hci_raw.c, 1025
 - ng_btsocket_hci_raw.h, 809
- ng_btsocket_hci_raw_send_ngmsg
 - ng_btsocket_hci_raw.c, 1026
- ng_btsocket_hci_raw_send_sync_ngmsg
 - ng_btsocket_hci_raw.c, 1026
- NG_BT_SOCKET_HCI_RAW_SENDSPACE
 - ng_btsocket_hci_raw.h, 806
- ng_btsocket_hci_raw_sockaddr
 - ng_btsocket_hci_raw.c, 1026
 - ng_btsocket_hci_raw.h, 810
- ng_btsocket_hci_raw_task
 - ng_btsocket_hci_raw.c, 1028
- ng_btsocket_hci_raw_usrreqs
 - ng_btsocket.c, 1013
- ng_btsocket_hci_raw_wakeup_input_task
 - ng_btsocket_hci_raw.c, 1019
- NG_BT_SOCKET_HCI_RAW_WARN
 - ng_btsocket_hci_raw.c, 1020
- NG_BT_SOCKET_INFO_LEVEL
 - ng_btsocket.h, 797
- ng_btsocket_l2cap.c
 - ifqmaxlen, 1048
 - LIST_HEAD, 1034
 - M_NETGRAPH_BT_SOCKET_L2CAP, 1032
 - ng_btsocket_l2cap_abort, 1034
 - ng_btsocket_l2cap_accept, 1034
 - NG_BT_SOCKET_L2CAP_ALERT, 1032
 - ng_btsocket_l2cap_attach, 1034

- ng_btsocket_l2cap_bind, 1034
- ng_btsocket_l2cap_close, 1034
- ng_btsocket_l2cap_connect, 1035
- ng_btsocket_l2cap_control, 1035
- ng_btsocket_l2cap_ctloutput, 1035
- ng_btsocket_l2cap_data_input, 1035
- ng_btsocket_l2cap_debug_level, 1048
- ng_btsocket_l2cap_default_msg_input, 1036
- ng_btsocket_l2cap_detach, 1036
- ng_btsocket_l2cap_disconnect, 1036
- NG_BT_SOCKET_L2CAP_ERR, 1033
- NG_BT_SOCKET_L2CAP_INFO, 1033
- ng_btsocket_l2cap_init, 1037
- ng_btsocket_l2cap_input, 1037
- ng_btsocket_l2cap_l2ca_msg_input, 1038
- ng_btsocket_l2cap_listen, 1038
- ng_btsocket_l2cap_node, 1048
- ng_btsocket_l2cap_node_connect, 1038, 1048
- ng_btsocket_l2cap_node_constructor, 1048
- ng_btsocket_l2cap_node_disconnect, 1039, 1048
- ng_btsocket_l2cap_node_newhook, 1039, 1049
- ng_btsocket_l2cap_node_rcvdata, 1039, 1049
- ng_btsocket_l2cap_node_rcvmsg, 1039, 1049
- ng_btsocket_l2cap_node_shutdown, 1039, 1049
- ng_btsocket_l2cap_pcb_by_addr, 1039
- ng_btsocket_l2cap_pcb_by_cid, 1040
- ng_btsocket_l2cap_pcb_by_token, 1040
- ng_btsocket_l2cap_peeraddr, 1040
- ng_btsocket_l2cap_process_l2ca_cfg_ind, 1040
- ng_btsocket_l2cap_process_l2ca_cfg_req_rsp, 1041
- ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp, 1041
- ng_btsocket_l2cap_process_l2ca_con_ind, 1042
- ng_btsocket_l2cap_process_l2ca_con_req_-rsp, 1042
- ng_btsocket_l2cap_process_l2ca_con_rsp_-rsp, 1043
- ng_btsocket_l2cap_process_l2ca_discon_ind, 1043
- ng_btsocket_l2cap_process_l2ca_discon_rsp, 1044
- ng_btsocket_l2cap_process_l2ca_write_rsp, 1044
- ng_btsocket_l2cap_process_timeout, 1044
- ng_btsocket_l2cap_queue, 1049
- ng_btsocket_l2cap_queue_mtx, 1049
- ng_btsocket_l2cap_queue_task, 1049
- ng_btsocket_l2cap_result2errno, 1045
- ng_btsocket_l2cap_rtclean, 1045
- ng_btsocket_l2cap_send, 1045
- ng_btsocket_l2cap_send2, 1046
- ng_btsocket_l2cap_send_l2ca_cfg_req, 1046
- ng_btsocket_l2cap_send_l2ca_cfg_rsp, 1046
- ng_btsocket_l2cap_send_l2ca_con_req, 1046
- ng_btsocket_l2cap_send_l2ca_con_rsp_req, 1047
- ng_btsocket_l2cap_send_l2ca_discon_req, 1047
- ng_btsocket_l2cap_sockaddr, 1047
- ng_btsocket_l2cap_timeout, 1047
- ng_btsocket_l2cap_untimeout, 1048
- ng_btsocket_l2cap_wakeup_input_task, 1033
- ng_btsocket_l2cap_wakeup_route_task, 1033
- NG_BT_SOCKET_L2CAP_WARN, 1033
- typestruct, 1049
- ng_btsocket_l2cap.h
 - ng_btsocket_l2cap_abort, 816
 - ng_btsocket_l2cap_accept, 816
 - ng_btsocket_l2cap_attach, 816
 - ng_btsocket_l2cap_bind, 816
 - NG_BT_SOCKET_L2CAP_CFG_BOTH, 812
 - NG_BT_SOCKET_L2CAP_CFG_IN, 812
 - NG_BT_SOCKET_L2CAP_CFG_IN_SENT, 812
 - NG_BT_SOCKET_L2CAP_CFG_OUT, 813
 - NG_BT_SOCKET_L2CAP_CFG_OUT_-SENT, 813
 - NG_BT_SOCKET_L2CAP_CLIENT, 813
 - ng_btsocket_l2cap_close, 816
 - NG_BT_SOCKET_L2CAP_CLOSED, 813
 - NG_BT_SOCKET_L2CAP_CONFIGURING, 813
 - ng_btsocket_l2cap_connect, 817
 - NG_BT_SOCKET_L2CAP_CONNECTING, 813
 - ng_btsocket_l2cap_control, 817
 - ng_btsocket_l2cap_ctloutput, 817
 - ng_btsocket_l2cap_detach, 817
 - ng_btsocket_l2cap_disconnect, 818
 - NG_BT_SOCKET_L2CAP_-DISCONNECTING, 813
 - ng_btsocket_l2cap_init, 818
 - ng_btsocket_l2cap_listen, 818
 - NG_BT_SOCKET_L2CAP_OPEN, 814
 - ng_btsocket_l2cap_pcb_p, 815
 - ng_btsocket_l2cap_pcb_t, 815
 - ng_btsocket_l2cap_peeraddr, 819
 - ng_btsocket_l2cap_raw_abort, 819
 - ng_btsocket_l2cap_raw_attach, 819
 - ng_btsocket_l2cap_raw_bind, 819
 - ng_btsocket_l2cap_raw_close, 819
 - ng_btsocket_l2cap_raw_connect, 820

- ng_btsocket_l2cap_raw_control, 820
- ng_btsocket_l2cap_raw_detach, 820
- ng_btsocket_l2cap_raw_disconnect, 821
- ng_btsocket_l2cap_raw_init, 821
- ng_btsocket_l2cap_raw_pcb_p, 815
- ng_btsocket_l2cap_raw_pcb_t, 815
- ng_btsocket_l2cap_raw_peeraddr, 821
- NG_BTSTACK_L2CAP_RAW_-
PRIVILEGED, 814
- NG_BTSTACK_L2CAP_RAW_-
RECVSPACE, 814
- ng_btsocket_l2cap_raw_send, 821
- NG_BTSTACK_L2CAP_RAW_-
SENDSPACE, 814
- ng_btsocket_l2cap_raw_sockaddr, 821
- NG_BTSTACK_L2CAP_RECVSPACE,
814
- ng_btsocket_l2cap_rentry_p, 815
- ng_btsocket_l2cap_rentry_t, 815
- ng_btsocket_l2cap_send, 822
- NG_BTSTACK_L2CAP_SENDSPACE, 814
- ng_btsocket_l2cap_sockaddr, 822
- NG_BTSTACK_L2CAP_TIMO, 814
- so2l2cap_pcb, 814
- so2l2cap_raw_pcb, 815
- ng_btsocket_l2cap_abort
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap.h, 816
- ng_btsocket_l2cap_accept
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap.h, 816
- NG_BTSTACK_L2CAP_ALERT
 - ng_btsocket_l2cap.c, 1032
- ng_btsocket_l2cap_attach
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap.h, 816
- ng_btsocket_l2cap_bind
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap.h, 816
- NG_BTSTACK_L2CAP_CFG_BOTH
 - ng_btsocket_l2cap.h, 812
- NG_BTSTACK_L2CAP_CFG_IN
 - ng_btsocket_l2cap.h, 812
- NG_BTSTACK_L2CAP_CFG_IN_SENT
 - ng_btsocket_l2cap.h, 812
- NG_BTSTACK_L2CAP_CFG_OUT
 - ng_btsocket_l2cap.h, 813
- NG_BTSTACK_L2CAP_CFG_OUT_SENT
 - ng_btsocket_l2cap.h, 813
- NG_BTSTACK_L2CAP_CLIENT
 - ng_btsocket_l2cap.h, 813
- ng_btsocket_l2cap_close
 - ng_btsocket_l2cap.c, 1034
 - ng_btsocket_l2cap.h, 816
- NG_BTSTACK_L2CAP_CLOSED
 - ng_btsocket_l2cap.h, 813
- NG_BTSTACK_L2CAP_CONFIGURING
 - ng_btsocket_l2cap.h, 813
- ng_btsocket_l2cap_connect
 - ng_btsocket_l2cap.c, 1035
 - ng_btsocket_l2cap.h, 817
- NG_BTSTACK_L2CAP_CONNECTING
 - ng_btsocket_l2cap.h, 813
- ng_btsocket_l2cap_control
 - ng_btsocket_l2cap.c, 1035
 - ng_btsocket_l2cap.h, 817
- ng_btsocket_l2cap_ctloutput
 - ng_btsocket_l2cap.c, 1035
 - ng_btsocket_l2cap.h, 817
- ng_btsocket_l2cap_data_input
 - ng_btsocket_l2cap.c, 1035
- ng_btsocket_l2cap_debug_level
 - ng_btsocket_l2cap.c, 1048
- ng_btsocket_l2cap_default_msg_input
 - ng_btsocket_l2cap.c, 1036
- ng_btsocket_l2cap_detach
 - ng_btsocket_l2cap.c, 1036
 - ng_btsocket_l2cap.h, 817
- ng_btsocket_l2cap_disconnect
 - ng_btsocket_l2cap.c, 1036
 - ng_btsocket_l2cap.h, 818
- NG_BTSTACK_L2CAP_DISCONNECTING
 - ng_btsocket_l2cap.h, 813
- NG_BTSTACK_L2CAP_ERR
 - ng_btsocket_l2cap.c, 1033
- NG_BTSTACK_L2CAP_INFO
 - ng_btsocket_l2cap.c, 1033
- ng_btsocket_l2cap_init
 - ng_btsocket_l2cap.c, 1037
 - ng_btsocket_l2cap.h, 818
- ng_btsocket_l2cap_input
 - ng_btsocket_l2cap.c, 1037
- ng_btsocket_l2cap_l2ca_msg_input
 - ng_btsocket_l2cap.c, 1038
- ng_btsocket_l2cap_listen
 - ng_btsocket_l2cap.c, 1038
 - ng_btsocket_l2cap.h, 818
- ng_btsocket_l2cap_node
 - ng_btsocket_l2cap.c, 1048
- ng_btsocket_l2cap_node_connect
 - ng_btsocket_l2cap.c, 1038, 1048
- ng_btsocket_l2cap_node_constructor
 - ng_btsocket_l2cap.c, 1048
- ng_btsocket_l2cap_node_disconnect
 - ng_btsocket_l2cap.c, 1039, 1048
- ng_btsocket_l2cap_node_newhook
 - ng_btsocket_l2cap.c, 1039, 1049
- ng_btsocket_l2cap_node_rcvdata

- ng_btsocket_l2cap.c, 1039, 1049
- ng_btsocket_l2cap_node_rcvmsg
 - ng_btsocket_l2cap.c, 1039, 1049
- ng_btsocket_l2cap_node_shutdown
 - ng_btsocket_l2cap.c, 1039, 1049
- NG_BT_SOCKET_L2CAP_NODE_TYPE
 - ng_btsocket.h, 797
- NG_BT_SOCKET_L2CAP_OPEN
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_pcb, 163
 - cfg_state, 164
 - cid, 164
 - dst, 164
 - flags, 164
 - flush_timo, 164
 - iflow, 165
 - imtu, 165
 - link_timo, 165
 - LIST_ENTRY, 164
 - oflow, 165
 - omtu, 165
 - pcb_mtx, 165
 - psm, 165
 - rt, 166
 - so, 166
 - src, 166
 - state, 166
 - timo, 167
 - token, 167
- ng_btsocket_l2cap_pcb_by_addr
 - ng_btsocket_l2cap.c, 1039
- ng_btsocket_l2cap_pcb_by_cid
 - ng_btsocket_l2cap.c, 1040
- ng_btsocket_l2cap_pcb_by_token
 - ng_btsocket_l2cap.c, 1040
- ng_btsocket_l2cap_pcb_p
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_pcb_t
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_peeraddr
 - ng_btsocket_l2cap.c, 1040
 - ng_btsocket_l2cap.h, 819
- ng_btsocket_l2cap_process_l2ca_cfg_ind
 - ng_btsocket_l2cap.c, 1040
- ng_btsocket_l2cap_process_l2ca_cfg_req_rsp
 - ng_btsocket_l2cap.c, 1041
- ng_btsocket_l2cap_process_l2ca_cfg_rsp_rsp
 - ng_btsocket_l2cap.c, 1041
- ng_btsocket_l2cap_process_l2ca_con_ind
 - ng_btsocket_l2cap.c, 1042
- ng_btsocket_l2cap_process_l2ca_con_req_rsp
 - ng_btsocket_l2cap.c, 1042
- ng_btsocket_l2cap_process_l2ca_con_rsp_rsp
 - ng_btsocket_l2cap.c, 1043
- ng_btsocket_l2cap_process_l2ca_discon_ind
 - ng_btsocket_l2cap.c, 1043
- ng_btsocket_l2cap_process_l2ca_discon_rsp
 - ng_btsocket_l2cap.c, 1044
- ng_btsocket_l2cap_process_l2ca_write_rsp
 - ng_btsocket_l2cap.c, 1044
- ng_btsocket_l2cap_process_timeout
 - ng_btsocket_l2cap.c, 1044
- ng_btsocket_l2cap_queue
 - ng_btsocket_l2cap.c, 1049
- ng_btsocket_l2cap_queue_mtx
 - ng_btsocket_l2cap.c, 1049
- ng_btsocket_l2cap_queue_task
 - ng_btsocket_l2cap.c, 1049
- ng_btsocket_l2cap_raw.c
 - ifqmaxlen, 1060
 - LIST_HEAD, 1055
 - M_NETGRAPH_BT_SOCKET_L2CAP_ -
RAW, 1054
 - ng_btsocket_l2cap_raw_abort, 1055
 - NG_BT_SOCKET_L2CAP_RAW_ALERT,
1054
 - ng_btsocket_l2cap_raw_attach, 1055
 - ng_btsocket_l2cap_raw_bind, 1055
 - ng_btsocket_l2cap_raw_close, 1055
 - ng_btsocket_l2cap_raw_connect, 1056
 - ng_btsocket_l2cap_raw_control, 1056
 - ng_btsocket_l2cap_raw_debug_level, 1060
 - ng_btsocket_l2cap_raw_detach, 1057
 - ng_btsocket_l2cap_raw_disconnect, 1057
 - NG_BT_SOCKET_L2CAP_RAW_ERR, 1054
 - ng_btsocket_l2cap_raw_get_token, 1057
 - NG_BT_SOCKET_L2CAP_RAW_INFO, 1054
 - ng_btsocket_l2cap_raw_init, 1057
 - ng_btsocket_l2cap_raw_input, 1057
 - ng_btsocket_l2cap_raw_ioctl_timeout, 1060
 - ng_btsocket_l2cap_raw_node, 1060
 - ng_btsocket_l2cap_raw_node_connect, 1058,
1060
 - ng_btsocket_l2cap_raw_node_constructor,
1060
 - ng_btsocket_l2cap_raw_node_disconnect,
1058, 1061
 - ng_btsocket_l2cap_raw_node_newhook,
1058, 1061
 - ng_btsocket_l2cap_raw_node_rcvdata, 1058,
1061
 - ng_btsocket_l2cap_raw_node_rcvmsg, 1058,
1061
 - ng_btsocket_l2cap_raw_node_shutdown,
1058, 1061
 - ng_btsocket_l2cap_raw_peeraddr, 1059
 - ng_btsocket_l2cap_raw_queue, 1061
 - ng_btsocket_l2cap_raw_queue_mtx, 1061

- ng_btsocket_l2cap_raw_queue_task, 1061
- ng_btsocket_l2cap_raw_rtclean, 1059
- ng_btsocket_l2cap_raw_send, 1059
- ng_btsocket_l2cap_raw_send_ngmsg, 1059
- ng_btsocket_l2cap_raw_send_sync_ngmsg, 1059
- ng_btsocket_l2cap_raw_sockaddr, 1060
- ng_btsocket_l2cap_raw_wakeup_input_task, 1054
- ng_btsocket_l2cap_raw_wakeup_route_task, 1054
- NG_BTSTACK_L2CAP_RAW_WARN, 1054
- typestruct, 1061
- ng_btsocket_l2cap_raw_abort
 - ng_btsocket_l2cap.h, 819
 - ng_btsocket_l2cap_raw.c, 1055
- NG_BTSTACK_L2CAP_RAW_ALERT
 - ng_btsocket_l2cap_raw.c, 1054
- ng_btsocket_l2cap_raw_attach
 - ng_btsocket_l2cap.h, 819
 - ng_btsocket_l2cap_raw.c, 1055
- ng_btsocket_l2cap_raw_auto_discon_timo, 168
- timeout, 168
- ng_btsocket_l2cap_raw_bind
 - ng_btsocket_l2cap.h, 819
 - ng_btsocket_l2cap_raw.c, 1055
- ng_btsocket_l2cap_raw_chan_list, 169
- channels, 169
- num_channels, 169
- ng_btsocket_l2cap_raw_close
 - ng_btsocket_l2cap.h, 819
 - ng_btsocket_l2cap_raw.c, 1055
- ng_btsocket_l2cap_raw_con_list, 170
- connections, 170
- num_connections, 170
- ng_btsocket_l2cap_raw_connect
 - ng_btsocket_l2cap.h, 820
 - ng_btsocket_l2cap_raw.c, 1056
- ng_btsocket_l2cap_raw_control
 - ng_btsocket_l2cap.h, 820
 - ng_btsocket_l2cap_raw.c, 1056
- ng_btsocket_l2cap_raw_debug_level
 - ng_btsocket_l2cap_raw.c, 1060
- ng_btsocket_l2cap_raw_detach
 - ng_btsocket_l2cap.h, 820
 - ng_btsocket_l2cap_raw.c, 1057
- ng_btsocket_l2cap_raw_disconnect
 - ng_btsocket_l2cap.h, 821
 - ng_btsocket_l2cap_raw.c, 1057
- NG_BTSTACK_L2CAP_RAW_ERR
 - ng_btsocket_l2cap_raw.c, 1054
- ng_btsocket_l2cap_raw_get_info, 171
- info_data, 171
- info_size, 171
- info_type, 171
- result, 171
- ng_btsocket_l2cap_raw_get_token
 - ng_btsocket_l2cap_raw.c, 1057
- NG_BTSTACK_L2CAP_RAW_INFO
 - ng_btsocket_l2cap_raw.c, 1054
- ng_btsocket_l2cap_raw_init
 - ng_btsocket_l2cap.h, 821
 - ng_btsocket_l2cap_raw.c, 1057
- ng_btsocket_l2cap_raw_input
 - ng_btsocket_l2cap_raw.c, 1057
- ng_btsocket_l2cap_raw_ioctl_timeout
 - ng_btsocket_l2cap_raw.c, 1060
- ng_btsocket_l2cap_raw_node
 - ng_btsocket_l2cap_raw.c, 1060
- ng_btsocket_l2cap_raw_node_connect
 - ng_btsocket_l2cap_raw.c, 1058, 1060
- ng_btsocket_l2cap_raw_node_constructor
 - ng_btsocket_l2cap_raw.c, 1060
- ng_btsocket_l2cap_raw_node_debug, 172
- debug, 172
- ng_btsocket_l2cap_raw_node_disconnect
 - ng_btsocket_l2cap_raw.c, 1058, 1061
- ng_btsocket_l2cap_raw_node_flags, 173
- flags, 173
- ng_btsocket_l2cap_raw_node_newhook
 - ng_btsocket_l2cap_raw.c, 1058, 1061
- ng_btsocket_l2cap_raw_node_rcvdata
 - ng_btsocket_l2cap_raw.c, 1058, 1061
- ng_btsocket_l2cap_raw_node_rcvmsg
 - ng_btsocket_l2cap_raw.c, 1058, 1061
- ng_btsocket_l2cap_raw_node_shutdown
 - ng_btsocket_l2cap_raw.c, 1058, 1061
- NG_BTSTACK_L2CAP_RAW_NODE_TYPE
 - ng_btsocket.h, 797
- ng_btsocket_l2cap_raw_pcb, 174
- dst, 174
- flags, 175
- LIST_ENTRY, 174
- msg, 175
- pcb_mtx, 175
- rt, 175
- so, 175
- src, 175
- token, 175
- ng_btsocket_l2cap_raw_pcb_p
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_raw_pcb_t
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_raw_peeraddr
 - ng_btsocket_l2cap.h, 821
 - ng_btsocket_l2cap_raw.c, 1059
- ng_btsocket_l2cap_raw_ping, 177

- echo_data, 177
- echo_size, 177
- result, 177
- NG_BT_SOCKET_L2CAP_RAW_PRIVILEGED
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_raw_queue
 - ng_btsocket_l2cap_raw.c, 1061
- ng_btsocket_l2cap_raw_queue_mtx
 - ng_btsocket_l2cap_raw.c, 1061
- ng_btsocket_l2cap_raw_queue_task
 - ng_btsocket_l2cap_raw.c, 1061
- NG_BT_SOCKET_L2CAP_RAW_RECVSPACE
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_raw_rtclean
 - ng_btsocket_l2cap_raw.c, 1059
- ng_btsocket_l2cap_raw_send
 - ng_btsocket_l2cap.h, 821
 - ng_btsocket_l2cap_raw.c, 1059
- ng_btsocket_l2cap_raw_send_ngmsg
 - ng_btsocket_l2cap_raw.c, 1059
- ng_btsocket_l2cap_raw_send_sync_ngmsg
 - ng_btsocket_l2cap_raw.c, 1059
- NG_BT_SOCKET_L2CAP_RAW_SENDSPACE
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_raw_sockaddr
 - ng_btsocket_l2cap.h, 821
 - ng_btsocket_l2cap_raw.c, 1060
- ng_btsocket_l2cap_raw_usrreqs
 - ng_btsocket.c, 1013
- ng_btsocket_l2cap_raw_wakeup_input_task
 - ng_btsocket_l2cap_raw.c, 1054
- ng_btsocket_l2cap_raw_wakeup_route_task
 - ng_btsocket_l2cap_raw.c, 1054
- NG_BT_SOCKET_L2CAP_RAW_WARN
 - ng_btsocket_l2cap_raw.c, 1054
- NG_BT_SOCKET_L2CAP_RECVSPACE
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_result2errno
 - ng_btsocket_l2cap.c, 1045
- ng_btsocket_l2cap_rtclean
 - ng_btsocket_l2cap.c, 1045
- ng_btsocket_l2cap_rtrentry, 178
 - hook, 178
 - LIST_ENTRY, 178
 - src, 178
- ng_btsocket_l2cap_rtrentry_p
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_rtrentry_t
 - ng_btsocket_l2cap.h, 815
- ng_btsocket_l2cap_send
 - ng_btsocket_l2cap.c, 1045
 - ng_btsocket_l2cap.h, 822
- ng_btsocket_l2cap_send2
 - ng_btsocket_l2cap.c, 1046
- ng_btsocket_l2cap_send_l2ca_cfg_req
 - ng_btsocket_l2cap.c, 1046
- ng_btsocket_l2cap_send_l2ca_cfg_rsp
 - ng_btsocket_l2cap.c, 1046
- ng_btsocket_l2cap_send_l2ca_con_req
 - ng_btsocket_l2cap.c, 1046
- ng_btsocket_l2cap_send_l2ca_con_rsp_req
 - ng_btsocket_l2cap.c, 1047
- ng_btsocket_l2cap_send_l2ca_discon_req
 - ng_btsocket_l2cap.c, 1047
- NG_BT_SOCKET_L2CAP_SENDSPACE
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_sockaddr
 - ng_btsocket_l2cap.c, 1047
 - ng_btsocket_l2cap.h, 822
- ng_btsocket_l2cap_timeout
 - ng_btsocket_l2cap.c, 1047
- NG_BT_SOCKET_L2CAP_TIMO
 - ng_btsocket_l2cap.h, 814
- ng_btsocket_l2cap_untimeout
 - ng_btsocket_l2cap.c, 1048
- ng_btsocket_l2cap_usrreqs
 - ng_btsocket.c, 1014
- ng_btsocket_l2cap_wakeup_input_task
 - ng_btsocket_l2cap.c, 1033
- ng_btsocket_l2cap_wakeup_route_task
 - ng_btsocket_l2cap.c, 1033
- NG_BT_SOCKET_L2CAP_WARN
 - ng_btsocket_l2cap.c, 1033
- ng_btsocket_mod
 - ng_btsocket.c, 1014
- ng_btsocket_modevent
 - ng_btsocket.c, 1012
- ng_btsocket_protosw
 - ng_btsocket.c, 1014
- ng_btsocket_protosw_end
 - ng_btsocket.c, 1012
- ng_btsocket_protosw_size
 - ng_btsocket.c, 1012
- ng_btsocket_rfcomm.c
 - ALOT, 1066
 - ifqmaxlen, 1088
 - LIST_HEAD, 1068
 - M_NETGRAPH_BT_SOCKET_RFCOMM, 1066
 - ng_btsocket_rfcomm_abort, 1068
 - ng_btsocket_rfcomm_accept, 1068
 - NG_BT_SOCKET_RFCOMM_ALERT, 1067
 - ng_btsocket_rfcomm_attach, 1068
 - ng_btsocket_rfcomm_bind, 1069
 - ng_btsocket_rfcomm_check_fcs, 1069
 - ng_btsocket_rfcomm_close, 1069
 - ng_btsocket_rfcomm_connect, 1069
 - ng_btsocket_rfcomm_connect_cfm, 1070

- ng_btsocket_rfcomm_connect_ind, 1070
- ng_btsocket_rfcomm_control, 1071
- ng_btsocket_rfcomm_crc, 1071
- ng_btsocket_rfcomm_ctloutput, 1071
- ng_btsocket_rfcomm_debug_level, 1088
- ng_btsocket_rfcomm_detach, 1071
- ng_btsocket_rfcomm_disconnect, 1072
- NG_BT_SOCKET_RFCOMM_ERR, 1067
- ng_btsocket_rfcomm_fcs2, 1072
- ng_btsocket_rfcomm_fcs3, 1072
- NG_BT_SOCKET_RFCOMM_INFO, 1067
- ng_btsocket_rfcomm_init, 1072
- ng_btsocket_rfcomm_listen, 1073
- ng_btsocket_rfcomm_pcb_by_channel, 1073
- ng_btsocket_rfcomm_pcb_by_dhci, 1073
- ng_btsocket_rfcomm_pcb_kill, 1073
- ng_btsocket_rfcomm_pcb_listener, 1074
- ng_btsocket_rfcomm_pcb_send, 1074
- ng_btsocket_rfcomm_peeraddr, 1074
- ng_btsocket_rfcomm_prepare_packet, 1075
- ng_btsocket_rfcomm_process_timeout, 1075
- ng_btsocket_rfcomm_receive_disc, 1075
- ng_btsocket_rfcomm_receive_dm, 1075
- ng_btsocket_rfcomm_receive_fc, 1076
- ng_btsocket_rfcomm_receive_frame, 1076
- ng_btsocket_rfcomm_receive_mcc, 1077
- ng_btsocket_rfcomm_receive_msc, 1077
- ng_btsocket_rfcomm_receive_pn, 1078
- ng_btsocket_rfcomm_receive_rls, 1078
- ng_btsocket_rfcomm_receive_rpn, 1079
- ng_btsocket_rfcomm_receive_sabm, 1079
- ng_btsocket_rfcomm_receive_test, 1080
- ng_btsocket_rfcomm_receive_ua, 1080
- ng_btsocket_rfcomm_receive_uih, 1081
- ng_btsocket_rfcomm_send, 1081
- ng_btsocket_rfcomm_send_command, 1081
- ng_btsocket_rfcomm_send_credits, 1082
- ng_btsocket_rfcomm_send_msc, 1082
- ng_btsocket_rfcomm_send_pn, 1083
- ng_btsocket_rfcomm_send_uih, 1083
- ng_btsocket_rfcomm_session_accept, 1083
- ng_btsocket_rfcomm_session_by_addr, 1084
- ng_btsocket_rfcomm_session_clean, 1084
- ng_btsocket_rfcomm_session_connect, 1084
- ng_btsocket_rfcomm_session_create, 1085
- ng_btsocket_rfcomm_session_process_pcb, 1085
- ng_btsocket_rfcomm_session_receive, 1086
- ng_btsocket_rfcomm_session_send, 1086
- ng_btsocket_rfcomm_session_task, 1086
- ng_btsocket_rfcomm_sessions_task, 1087
- ng_btsocket_rfcomm_set_pn, 1087
- ng_btsocket_rfcomm_sockaddr, 1087
- ng_btsocket_rfcomm_task, 1089
- ng_btsocket_rfcomm_task_wakeup, 1067
- ng_btsocket_rfcomm_timeout, 1088
- ng_btsocket_rfcomm_timo, 1089
- ng_btsocket_rfcomm_untimeout, 1088
- ng_btsocket_rfcomm_upcall, 1088
- NG_BT_SOCKET_RFCOMM_WARN, 1068
- ng_btsocket_rfcomm.h
 - INITIATOR, 826
 - ng_btsocket_rfcomm_abort, 838
 - ng_btsocket_rfcomm_accept, 838
 - ng_btsocket_rfcomm_attach, 838
 - ng_btsocket_rfcomm_bind, 839
 - ng_btsocket_rfcomm_close, 839
 - ng_btsocket_rfcomm_connect, 839
 - ng_btsocket_rfcomm_control, 840
 - ng_btsocket_rfcomm_ctloutput, 840
 - ng_btsocket_rfcomm_detach, 840
 - ng_btsocket_rfcomm_disconnect, 840
 - NG_BT_SOCKET_RFCOMM_DLC_CFC, 826
 - NG_BT_SOCKET_RFCOMM_DLC_CLOSED, 826
 - NG_BT_SOCKET_RFCOMM_DLC_CONFIGURING, 826
 - NG_BT_SOCKET_RFCOMM_DLC_CONNECTED, 826
 - NG_BT_SOCKET_RFCOMM_DLC_CONNECTING, 827
 - NG_BT_SOCKET_RFCOMM_DLC_DETACHED, 827
 - NG_BT_SOCKET_RFCOMM_DLC_DISCONNECTING, 827
 - NG_BT_SOCKET_RFCOMM_DLC_SENDING, 827
 - NG_BT_SOCKET_RFCOMM_DLC_TIMEDOUT, 827
 - NG_BT_SOCKET_RFCOMM_DLC_TIMO, 827
 - NG_BT_SOCKET_RFCOMM_DLC_W4_CONNECT, 827
 - ng_btsocket_rfcomm_init, 841
 - ng_btsocket_rfcomm_listen, 841
 - ng_btsocket_rfcomm_pcb_p, 838
 - ng_btsocket_rfcomm_pcb_t, 838
 - ng_btsocket_rfcomm_peeraddr, 841
 - NG_BT_SOCKET_RFCOMM_RECVSPACE, 828
 - ng_btsocket_rfcomm_send, 841
 - NG_BT_SOCKET_RFCOMM_SENDSPACE, 828
 - NG_BT_SOCKET_RFCOMM_SESSION_CLOSED, 828
 - NG_BT_SOCKET_RFCOMM_SESSION_CONNECTED, 828

- NG_BT_SOCKET_RFCOMM_SESSION_-CONNECTING, 828
- NG_BT_SOCKET_RFCOMM_SESSION_-DISCONNECTING, 828
- NG_BT_SOCKET_RFCOMM_SESSION_-INITIATOR, 828
- NG_BT_SOCKET_RFCOMM_SESSION_-LFC, 829
- NG_BT_SOCKET_RFCOMM_SESSION_-LISTENING, 829
- NG_BT_SOCKET_RFCOMM_SESSION_-OPEN, 829
- ng_btsocket_rfcomm_session_p, 838
- NG_BT_SOCKET_RFCOMM_SESSION_-RFC, 829
- ng_btsocket_rfcomm_session_t, 838
- ng_btsocket_rfcomm_sockaddr, 842
- packed, 842
- RFCOMM_CHANNEL, 829
- RFCOMM_CR, 829
- RFCOMM_DEFAULT_CREDITS, 829
- RFCOMM_DEFAULT_MTU, 829
- RFCOMM_DIRECTION, 830
- RFCOMM_DLCI, 830
- RFCOMM_EA, 830
- RFCOMM_FRAME_DISC, 830
- RFCOMM_FRAME_DM, 830
- RFCOMM_FRAME_SABM, 830
- RFCOMM_FRAME_UA, 830
- RFCOMM_FRAME_UIH, 830
- RFCOMM_MAX_CREDITS, 831
- RFCOMM_MAX_MTU, 831
- RFCOMM_MCC_FCOFF, 831
- RFCOMM_MCC_FCON, 831
- RFCOMM_MCC_LENGTH, 831
- RFCOMM_MCC_MSC, 831
- RFCOMM_MCC_NSC, 831
- RFCOMM_MCC_PN, 831
- RFCOMM_MCC_RLS, 832
- RFCOMM_MCC_RPN, 832
- RFCOMM_MCC_TEST, 832
- RFCOMM_MCC_TYPE, 832
- RFCOMM_MKADDRESS, 832
- RFCOMM_MKCONTROL, 832
- RFCOMM_MKDLCI, 832
- RFCOMM_MKLEN16, 832
- RFCOMM_MKLEN8, 833
- RFCOMM_MKMCC_TYPE, 833
- RFCOMM_MKRPN_LINE_SETTINGS, 833
- RFCOMM_MODEM_DV, 833
- RFCOMM_MODEM_FC, 833
- RFCOMM_MODEM_IC, 833
- RFCOMM_MODEM_RTC, 833
- RFCOMM_MODEM_RTR, 833
- RFCOMM_PF, 834
- RFCOMM_RPN_BR_115200, 834
- RFCOMM_RPN_BR_19200, 834
- RFCOMM_RPN_BR_230400, 834
- RFCOMM_RPN_BR_2400, 834
- RFCOMM_RPN_BR_38400, 834
- RFCOMM_RPN_BR_4800, 834
- RFCOMM_RPN_BR_57600, 834
- RFCOMM_RPN_BR_7200, 834
- RFCOMM_RPN_BR_9600, 834
- RFCOMM_RPN_DATA_5, 835
- RFCOMM_RPN_DATA_6, 835
- RFCOMM_RPN_DATA_7, 835
- RFCOMM_RPN_DATA_8, 835
- RFCOMM_RPN_DATA_BITS, 835
- RFCOMM_RPN_FLOW_NONE, 835
- RFCOMM_RPN_PARITY, 835
- RFCOMM_RPN_PARITY_EVEN, 835
- RFCOMM_RPN_PARITY_MARK, 835
- RFCOMM_RPN_PARITY_NONE, 835
- RFCOMM_RPN_PARITY_ODD, 836
- RFCOMM_RPN_PARITY_SPACE, 836
- RFCOMM_RPN_PM_ALL, 836
- RFCOMM_RPN_PM_BITRATE, 836
- RFCOMM_RPN_PM_DATA, 836
- RFCOMM_RPN_PM_FLOW, 836
- RFCOMM_RPN_PM_PARITY, 836
- RFCOMM_RPN_PM_PARITY_TYPE, 836
- RFCOMM_RPN_PM_STOP, 836
- RFCOMM_RPN_PM_XOFF, 836
- RFCOMM_RPN_PM_XON, 837
- RFCOMM_RPN_STOP_1, 837
- RFCOMM_RPN_STOP_15, 837
- RFCOMM_RPN_STOP_BITS, 837
- RFCOMM_RPN_XOFF_CHAR, 837
- RFCOMM_RPN_XON_CHAR, 837
- RFCOMM_SRVCHANNEL, 837
- RFCOMM_TYPE, 837
- so2rfcomm_pcb, 837
- ng_btsocket_rfcomm_abort
 - ng_btsocket_rfcomm.c, 1068
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_accept
 - ng_btsocket_rfcomm.c, 1068
 - ng_btsocket_rfcomm.h, 838
- NG_BT_SOCKET_RFCOMM_ALERT
 - ng_btsocket_rfcomm.c, 1067
- ng_btsocket_rfcomm_attach
 - ng_btsocket_rfcomm.c, 1068
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_bind
 - ng_btsocket_rfcomm.c, 1069
 - ng_btsocket_rfcomm.h, 839
- ng_btsocket_rfcomm_check_fcs

- ng_btsocket_rfcomm.c, 1069
- ng_btsocket_rfcomm_close
 - ng_btsocket_rfcomm.c, 1069
 - ng_btsocket_rfcomm.h, 839
- ng_btsocket_rfcomm_connect
 - ng_btsocket_rfcomm.c, 1069
 - ng_btsocket_rfcomm.h, 839
- ng_btsocket_rfcomm_connect_cfm
 - ng_btsocket_rfcomm.c, 1070
- ng_btsocket_rfcomm_connect_ind
 - ng_btsocket_rfcomm.c, 1070
- ng_btsocket_rfcomm_control
 - ng_btsocket_rfcomm.c, 1071
 - ng_btsocket_rfcomm.h, 840
- ng_btsocket_rfcomm_crc
 - ng_btsocket_rfcomm.c, 1071
- ng_btsocket_rfcomm_ctloutput
 - ng_btsocket_rfcomm.c, 1071
 - ng_btsocket_rfcomm.h, 840
- ng_btsocket_rfcomm_debug_level
 - ng_btsocket_rfcomm.c, 1088
- ng_btsocket_rfcomm_detach
 - ng_btsocket_rfcomm.c, 1071
 - ng_btsocket_rfcomm.h, 840
- ng_btsocket_rfcomm_disconnect
 - ng_btsocket_rfcomm.c, 1072
 - ng_btsocket_rfcomm.h, 840
- NG_BTSTACK_RFCOMM_DLC_CFC
 - ng_btsocket_rfcomm.h, 826
- NG_BTSTACK_RFCOMM_DLC_CLOSED
 - ng_btsocket_rfcomm.h, 826
- NG_BTSTACK_RFCOMM_DLC_-
CONFIGURING
 - ng_btsocket_rfcomm.h, 826
- NG_BTSTACK_RFCOMM_DLC_-
CONNECTED
 - ng_btsocket_rfcomm.h, 826
- NG_BTSTACK_RFCOMM_DLC_-
CONNECTING
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_DETACHED
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_-
DISCONNECTING
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_SENDING
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_TIMEDOUT
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_TIMO
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_DLC_W4_-
CONNECT
 - ng_btsocket_rfcomm.h, 827
- NG_BTSTACK_RFCOMM_ERR
 - ng_btsocket_rfcomm.c, 1067
- ng_btsocket_rfcomm_fc_info, 180
 - cfc, 180
 - lmodem, 180
 - reserved, 180
 - rmodem, 180
 - rx_cred, 180
 - tx_cred, 180
- ng_btsocket_rfcomm_fcs2
 - ng_btsocket_rfcomm.c, 1072
- ng_btsocket_rfcomm_fcs3
 - ng_btsocket_rfcomm.c, 1072
- NG_BTSTACK_RFCOMM_INFO
 - ng_btsocket_rfcomm.c, 1067
- ng_btsocket_rfcomm_init
 - ng_btsocket_rfcomm.c, 1072
 - ng_btsocket_rfcomm.h, 841
- ng_btsocket_rfcomm_listen
 - ng_btsocket_rfcomm.c, 1073
 - ng_btsocket_rfcomm.h, 841
- ng_btsocket_rfcomm_pcb, 182
 - channel, 183
 - dlsi, 183
 - dst, 183
 - flags, 183
 - LIST_ENTRY, 183
 - lmodem, 183
 - mtu, 183
 - pcb_mtx, 184
 - rmodem, 184
 - rx_cred, 184
 - session, 184
 - so, 184
 - src, 184
 - state, 185
 - timo, 185
 - tx_cred, 185
- ng_btsocket_rfcomm_pcb_by_channel
 - ng_btsocket_rfcomm.c, 1073
- ng_btsocket_rfcomm_pcb_by_dlsi
 - ng_btsocket_rfcomm.c, 1073
- ng_btsocket_rfcomm_pcb_kill
 - ng_btsocket_rfcomm.c, 1073
- ng_btsocket_rfcomm_pcb_listener
 - ng_btsocket_rfcomm.c, 1074
- ng_btsocket_rfcomm_pcb_p
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_pcb_send
 - ng_btsocket_rfcomm.c, 1074
- ng_btsocket_rfcomm_pcb_t
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_peeraddr
 - ng_btsocket_rfcomm.c, 1074

- ng_btsocket_rfcomm.h, 841
- ng_btsocket_rfcomm_prepare_packet
 - ng_btsocket_rfcomm.c, 1075
- ng_btsocket_rfcomm_process_timeout
 - ng_btsocket_rfcomm.c, 1075
- ng_btsocket_rfcomm_receive_disc
 - ng_btsocket_rfcomm.c, 1075
- ng_btsocket_rfcomm_receive_dm
 - ng_btsocket_rfcomm.c, 1075
- ng_btsocket_rfcomm_receive_fc
 - ng_btsocket_rfcomm.c, 1076
- ng_btsocket_rfcomm_receive_frame
 - ng_btsocket_rfcomm.c, 1076
- ng_btsocket_rfcomm_receive_mcc
 - ng_btsocket_rfcomm.c, 1077
- ng_btsocket_rfcomm_receive_msc
 - ng_btsocket_rfcomm.c, 1077
- ng_btsocket_rfcomm_receive_pn
 - ng_btsocket_rfcomm.c, 1078
- ng_btsocket_rfcomm_receive_rls
 - ng_btsocket_rfcomm.c, 1078
- ng_btsocket_rfcomm_receive_rpn
 - ng_btsocket_rfcomm.c, 1079
- ng_btsocket_rfcomm_receive_sabm
 - ng_btsocket_rfcomm.c, 1079
- ng_btsocket_rfcomm_receive_test
 - ng_btsocket_rfcomm.c, 1080
- ng_btsocket_rfcomm_receive_ua
 - ng_btsocket_rfcomm.c, 1080
- ng_btsocket_rfcomm_receive_uih
 - ng_btsocket_rfcomm.c, 1081
- NG_BTSTACK_RFCOMM_RECVSPACE
 - ng_btsocket_rfcomm.h, 828
- ng_btsocket_rfcomm_send
 - ng_btsocket_rfcomm.c, 1081
 - ng_btsocket_rfcomm.h, 841
- ng_btsocket_rfcomm_send_command
 - ng_btsocket_rfcomm.c, 1081
- ng_btsocket_rfcomm_send_credits
 - ng_btsocket_rfcomm.c, 1082
- ng_btsocket_rfcomm_send_msc
 - ng_btsocket_rfcomm.c, 1082
- ng_btsocket_rfcomm_send_pn
 - ng_btsocket_rfcomm.c, 1083
- ng_btsocket_rfcomm_send_uih
 - ng_btsocket_rfcomm.c, 1083
- NG_BTSTACK_RFCOMM_SENDSPACE
 - ng_btsocket_rfcomm.h, 828
- ng_btsocket_rfcomm_session, 186
 - flags, 186
 - l2so, 186
 - LIST_ENTRY, 186
 - LIST_HEAD, 186
 - mtu, 187
 - outq, 187
 - session_mtx, 187
 - state, 187
- ng_btsocket_rfcomm_session_accept
 - ng_btsocket_rfcomm.c, 1083
- ng_btsocket_rfcomm_session_by_addr
 - ng_btsocket_rfcomm.c, 1084
- ng_btsocket_rfcomm_session_clean
 - ng_btsocket_rfcomm.c, 1084
- NG_BTSTACK_RFCOMM_SESSION_CLOSED
 - ng_btsocket_rfcomm.h, 828
- ng_btsocket_rfcomm_session_connect
 - ng_btsocket_rfcomm.c, 1084
- NG_BTSTACK_RFCOMM_SESSION_CONNECTED
 - ng_btsocket_rfcomm.h, 828
- NG_BTSTACK_RFCOMM_SESSION_CONNECTING
 - ng_btsocket_rfcomm.h, 828
- ng_btsocket_rfcomm_session_create
 - ng_btsocket_rfcomm.c, 1085
- NG_BTSTACK_RFCOMM_SESSION_DISCONNECTING
 - ng_btsocket_rfcomm.h, 828
- NG_BTSTACK_RFCOMM_SESSION_INITIATOR
 - ng_btsocket_rfcomm.h, 828
- NG_BTSTACK_RFCOMM_SESSION_LFC
 - ng_btsocket_rfcomm.h, 829
- NG_BTSTACK_RFCOMM_SESSION_LISTENING
 - ng_btsocket_rfcomm.h, 829
- NG_BTSTACK_RFCOMM_SESSION_OPEN
 - ng_btsocket_rfcomm.h, 829
- ng_btsocket_rfcomm_session_p
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_session_process_pcb
 - ng_btsocket_rfcomm.c, 1085
- ng_btsocket_rfcomm_session_receive
 - ng_btsocket_rfcomm.c, 1086
- NG_BTSTACK_RFCOMM_SESSION_RFC
 - ng_btsocket_rfcomm.h, 829
- ng_btsocket_rfcomm_session_send
 - ng_btsocket_rfcomm.c, 1086
- ng_btsocket_rfcomm_session_t
 - ng_btsocket_rfcomm.h, 838
- ng_btsocket_rfcomm_session_task
 - ng_btsocket_rfcomm.c, 1086
- ng_btsocket_rfcomm_sessions_task
 - ng_btsocket_rfcomm.c, 1087
- ng_btsocket_rfcomm_set_pn
 - ng_btsocket_rfcomm.c, 1087
- ng_btsocket_rfcomm_sockaddr

- ng_btsocket_rfcomm.c, 1087
- ng_btsocket_rfcomm.h, 842
- ng_btsocket_rfcomm_task
 - ng_btsocket_rfcomm.c, 1089
- ng_btsocket_rfcomm_task_wakeup
 - ng_btsocket_rfcomm.c, 1067
- ng_btsocket_rfcomm_timeout
 - ng_btsocket_rfcomm.c, 1088
- ng_btsocket_rfcomm_timo
 - ng_btsocket_rfcomm.c, 1089
- ng_btsocket_rfcomm_untimeout
 - ng_btsocket_rfcomm.c, 1088
- ng_btsocket_rfcomm_upcall
 - ng_btsocket_rfcomm.c, 1088
- ng_btsocket_rfcomm_usrreqs
 - ng_btsocket.c, 1014
- NG_BTsocketRFCOMM_WARN
 - ng_btsocket_rfcomm.c, 1068
- NG_BTsocket_WARN_LEVEL
 - ng_btsocket.h, 797
- ng_bypass
 - netgraph.h, 1151
 - ng_base.c, 1185
- ng_bytearray_getDefault
 - ng_parse.c, 1454
- ng_bytearray_parse
 - ng_parse.c, 1454
- ng_bytearray_unparse
 - ng_parse.c, 1454
- ng_callout
 - netgraph.h, 1152
 - ng_base.c, 1185
- ng_callout_init
 - netgraph.h, 1128
- ng_callout_trampoline
 - ng_base.c, 1186
- ng_ccatm.c
 - __FBSDDID, 555
 - MALLOC_DEFINE, 555
 - MODULE_DEPEND, 555
 - NETGRAPH_INIT, 555
 - ng_ccatm_addr_array_type, 560
 - ng_ccatm_addr_array_type_info, 560
 - ng_ccatm_addr_req_array_getlen, 555
 - ng_ccatm_addr_req_array_type, 560
 - ng_ccatm_addr_req_array_type_info, 560
 - ng_ccatm_addr_req_type, 560
 - ng_ccatm_addr_req_type_info, 560
 - ng_ccatm_atm_port_type, 561
 - ng_ccatm_atm_port_type_info, 561
 - ng_ccatm_cmdlist, 561
 - ng_ccatm_constructor, 555, 561
 - ng_ccatm_disconnect, 555, 561
 - ng_ccatm_dump, 555
 - ng_ccatm_esi_type, 561
 - ng_ccatm_esi_type_info, 561
 - ng_ccatm_get_addresses, 556
 - ng_ccatm_get_addresses_type, 562
 - ng_ccatm_get_addresses_type_info, 562
 - ng_ccatm_log, 556
 - ng_ccatm_mod_event, 556
 - ng_ccatm_newhook, 556, 562
 - ng_ccatm_port_array_getlen, 557
 - ng_ccatm_port_array_type, 562
 - ng_ccatm_port_array_type_info, 562
 - ng_ccatm_port_type, 562
 - ng_ccatm_port_type_info, 563
 - ng_ccatm_portlist_type, 563
 - ng_ccatm_portlist_type_info, 563
 - ng_ccatm_rcvdata, 557, 563
 - ng_ccatm_rcvdump, 557, 563
 - ng_ccatm_rcvmanage, 557, 563
 - ng_ccatm_rcvmsg, 557, 563
 - ng_ccatm_rcvuni, 558, 564
 - ng_ccatm_respond_user, 558
 - ng_ccatm_send_uni, 558
 - ng_ccatm_send_uni_glob, 558
 - ng_ccatm_send_user, 559
 - ng_ccatm_shutdown, 559, 564
 - ng_ccatm_tpestruct, 564
 - ng_ccatm_uni_addr_type, 564
 - ng_ccatm_uni_addr_type_info, 564
 - pack_buf, 559
 - send_dump, 559
- ng_ccatm.h
 - NGM_CCATM_ADDRESS_REGISTERED, 596
 - NGM_CCATM_ADDRESS_-UNREGISTERED, 596
 - NGM_CCATM_CLEAR, 596
 - NGM_CCATM_DUMP, 596
 - NGM_CCATM_GET_ADDRESSES, 596
 - NGM_CCATM_GET_EXSTAT, 596
 - NGM_CCATM_GET_PORT_PARAM, 596
 - NGM_CCATM_GET_PORTLIST, 596
 - NGM_CCATM_GETSTATE, 596
 - NGM_CCATM_RESET, 596
 - NGM_CCATM_SET_PORT_PARAM, 596
 - NGM_CCATM_SETLOG, 596
 - NGM_CCATM_START, 596
 - NGM_CCATM_STOP, 596
- ng_ccatm.h
 - NG_CCATM_NODE_TYPE, 593
 - NGM_CCATM_ADDR_ARRAY_INFO, 593
 - NGM_CCATM_ADDR_REQ_ARRAY_-INFO, 594
 - NGM_CCATM_ADDR_REQ_INFO, 594
 - NGM_CCATM_ATM_PORT_INFO, 594

- NGM_CCATM_COOKIE, 594
- NGM_CCATM_ESI_INFO, 595
- NGM_CCATM_GET_ADDRESSES_INFO, 595
- NGM_CCATM_PORT_ARRAY_INFO, 595
- NGM_CCATM_PORT_INFO, 595
- NGM_CCATM_PORTLIST_INFO, 595
- NGM_CCATM_UNI_ADDR_INFO, 596
- ng_ccatm_addr_array_type
 - ng_ccatm.c, 560
- ng_ccatm_addr_array_type_info
 - ng_ccatm.c, 560
- ng_ccatm_addr_req_array_getlen
 - ng_ccatm.c, 555
- ng_ccatm_addr_req_array_type
 - ng_ccatm.c, 560
- ng_ccatm_addr_req_array_type_info
 - ng_ccatm.c, 560
- ng_ccatm_addr_req_type
 - ng_ccatm.c, 560
- ng_ccatm_addr_req_type_info
 - ng_ccatm.c, 560
- ng_ccatm_atm_port_type
 - ng_ccatm.c, 561
- ng_ccatm_atm_port_type_info
 - ng_ccatm.c, 561
- ng_ccatm_cmdlist
 - ng_ccatm.c, 561
- ng_ccatm_constructor
 - ng_ccatm.c, 555, 561
- ng_ccatm_cust.h
 - CCASSERT, 566
 - CCFREE, 566
 - CCGETERRNO, 566
 - CCMALLOC, 566
 - CCZALLOC, 566
 - MALLOC_DECLARE, 566
- ng_ccatm_disconnect
 - ng_ccatm.c, 555, 561
- ng_ccatm_dump
 - ng_ccatm.c, 555
- ng_ccatm_esi_type
 - ng_ccatm.c, 561
- ng_ccatm_esi_type_info
 - ng_ccatm.c, 561
- ng_ccatm_get_addresses
 - ng_ccatm.c, 556
- ng_ccatm_get_addresses_type
 - ng_ccatm.c, 562
- ng_ccatm_get_addresses_type_info
 - ng_ccatm.c, 562
- ng_ccatm_log
 - ng_ccatm.c, 556
- ng_ccatm_mod_event
 - ng_ccatm.c, 556
- ng_ccatm_newhook
 - ng_ccatm.c, 556, 562
- NG_CCATM_NODE_TYPE
 - ng_ccatm.h, 593
- ng_ccatm_port_array_getlen
 - ng_ccatm.c, 557
- ng_ccatm_port_array_type
 - ng_ccatm.c, 562
- ng_ccatm_port_array_type_info
 - ng_ccatm.c, 562
- ng_ccatm_port_type
 - ng_ccatm.c, 562
- ng_ccatm_port_type_info
 - ng_ccatm.c, 563
- ng_ccatm_portlist_type
 - ng_ccatm.c, 563
- ng_ccatm_portlist_type_info
 - ng_ccatm.c, 563
- ng_ccatm_rcvdata
 - ng_ccatm.c, 557, 563
- ng_ccatm_rcvdump
 - ng_ccatm.c, 557, 563
- ng_ccatm_rcvmanage
 - ng_ccatm.c, 557, 563
- ng_ccatm_rcvmsg
 - ng_ccatm.c, 557, 563
- ng_ccatm_rcvuni
 - ng_ccatm.c, 558, 564
- ng_ccatm_respond_user
 - ng_ccatm.c, 558
- ng_ccatm_send_uni
 - ng_ccatm.c, 558
- ng_ccatm_send_uni_glob
 - ng_ccatm.c, 558
- ng_ccatm_send_user
 - ng_ccatm.c, 559
- ng_ccatm_shutdown
 - ng_ccatm.c, 559, 564
- ng_ccatm_typestruct
 - ng_ccatm.c, 564
- ng_ccatm_uni_addr_type
 - ng_ccatm.c, 564
- ng_ccatm_uni_addr_type_info
 - ng_ccatm.c, 564
- ng_cisco.c
 - CISCO_ADDR_REPLY, 1230
 - CISCO_ADDR_REQ, 1230
 - cisco_constructor, 1232, 1234
 - cisco_disconnect, 1232, 1234
 - CISCO_HEADER_LEN, 1230
 - cisco_input, 1232
 - CISCO_KEEPALIVE, 1231
 - cisco_keepalive, 1232

- CISCO_KEEPALIVE_REQ, 1231
- CISCO_MULTICAST, 1231
- cisco_newhook, 1233, 1234
- cisco_notify, 1233
- CISCO_PACKET_LEN, 1231
- cisco_rcvdata, 1233, 1234
- cisco_rcvmsg, 1233, 1234
- cisco_send, 1234
- cisco_shutdown, 1234
- CISCO_UNICAST, 1231
- KEEPALIVE_SECS, 1231
- NETGRAPH_INIT, 1234
- ng_cisco_cmdlist, 1235
- ng_cisco_ipaddr_type, 1235
- ng_cisco_ipaddr_type_fields, 1235
- ng_cisco_stats_type, 1235
- ng_cisco_stats_type_fields, 1236
- sc_p, 1231
- typestruct, 1236
- ng_cisco.h
 - NGM_CISCO_GET_IPADDR, 1239
 - NGM_CISCO_GET_STATUS, 1239
 - NGM_CISCO_SET_IPADDR, 1239
- ng_cisco.h
 - NG_CISCO_HOOK_APPLETALK, 1237
 - NG_CISCO_HOOK_DEBUG, 1237
 - NG_CISCO_HOOK_DOWNSTREAM, 1237
 - NG_CISCO_HOOK_INET, 1237
 - NG_CISCO_HOOK_INET6, 1238
 - NG_CISCO_HOOK_IPX, 1238
 - NG_CISCO_HOOK_IPX, 1238
 - NG_CISCO_IPADDR_TYPE_INFO, 1238
 - NG_CISCO_NODE_TYPE, 1238
 - NG_CISCO_STATS_TYPE_INFO, 1238
 - NGM_CISCO_COOKIE, 1238
- ng_cisco_cmdlist
 - ng_cisco.c, 1235
- NG_CISCO_HOOK_APPLETALK
 - ng_cisco.h, 1237
- NG_CISCO_HOOK_DEBUG
 - ng_cisco.h, 1237
- NG_CISCO_HOOK_DOWNSTREAM
 - ng_cisco.h, 1237
- NG_CISCO_HOOK_INET
 - ng_cisco.h, 1237
- NG_CISCO_HOOK_INET6
 - ng_cisco.h, 1238
- NG_CISCO_HOOK_IPX
 - ng_cisco.h, 1238
- ng_cisco_ipaddr, 189
 - ipaddr, 189
 - netmask, 189
- ng_cisco_ipaddr_type
 - ng_cisco.c, 1235
 - ng_iface.c, 1342
- ng_cisco_ipaddr_type_fields
 - ng_cisco.c, 1235
 - ng_iface.c, 1343
- NG_CISCO_IPADDR_TYPE_INFO
 - ng_cisco.h, 1238
- NG_CISCO_NODE_TYPE
 - ng_cisco.h, 1238
- ng_cisco_stats, 190
 - keepAlivePeriod, 190
 - seqRetries, 190
- ng_cisco_stats_type
 - ng_cisco.c, 1235
- ng_cisco_stats_type_fields
 - ng_cisco.c, 1236
- NG_CISCO_STATS_TYPE_INFO
 - ng_cisco.h, 1238
- ng_close_t
 - netgraph.h, 1150
- NG_CLOSING
 - netgraph.h, 1128
- ng_cmdlist, 191
 - cmd, 191
 - cookie, 191
 - msgType, 191
 - name, 191
 - respType, 191
- NG_CMDSTRLEN
 - ng_message.h, 1409
- NG_CMDSTRSIZ
 - ng_message.h, 1409
- ng_con_nodes
 - ng_base.c, 1186
- ng_con_part2
 - ng_base.c, 1186
- ng_con_part3
 - ng_base.c, 1187
- ng_connect_data
 - ng_socket.c, 1584
- ng_connect_t
 - netgraph.h, 1150
- ng_constructor_t
 - netgraph.h, 1150
- NG_CONTROL
 - ng_socket.h, 1593
- ng_copy_meta
 - netgraph.h, 1128
- NG_COPYMESSAGE
 - ng_message.h, 1409
- NG_DATA
 - ng_socket.h, 1593
- ng_deadhook
 - ng_base.c, 1201
- ng_deadnode
 - ng_base.c, 1201

- ng_decodeidname
 - ng_base.c, 1187
- ng_deflate.c
 - DEFLATE_BUF_SIZE, 1241
 - DEFLATE_HDRLEN, 1241
 - ERROUT, 1242
 - MALLOC_DEFINE, 1242
 - MODULE_DEPEND, 1242
 - NETGRAPH_INIT, 1242
 - ng_deflate_cmds, 1244
 - ng_deflate_compress, 1242
 - ng_deflate_config_type, 1244
 - ng_deflate_config_type_fields, 1244
 - ng_deflate_constructor, 1242, 1245
 - ng_deflate_decompress, 1242
 - ng_deflate_disconnect, 1243, 1245
 - ng_deflate_newhook, 1243, 1245
 - ng_deflate_rcvdata, 1243, 1245
 - ng_deflate_rcvmsg, 1243, 1245
 - ng_deflate_reset_req, 1244
 - ng_deflate_shutdown, 1244, 1245
 - ng_deflate_stat_type, 1245
 - ng_deflate_stats_type_fields, 1245
 - ng_deflate_tpestruct, 1245
 - priv_p, 1242
 - PROT_COMPD, 1242
 - z_alloc, 1244
 - z_free, 1244
- ng_deflate.h
 - NGM_DEFLATE_CLR_STATS, 1248
 - NGM_DEFLATE_CONFIG, 1248
 - NGM_DEFLATE_GET_STATS, 1248
 - NGM_DEFLATE_GETCLR_STATS, 1248
 - NGM_DEFLATE_RESETRREQ, 1248
- ng_deflate.h
 - NG_DEFLATE_CONFIG_INFO, 1247
 - NG_DEFLATE_HOOK_COMP, 1247
 - NG_DEFLATE_HOOK_DECOMP, 1247
 - NG_DEFLATE_NODE_TYPE, 1248
 - NG_DEFLATE_STATS_INFO, 1248
 - NGM_DEFLATE_COOKIE, 1248
- ng_deflate_cmds
 - ng_deflate.c, 1244
- ng_deflate_compress
 - ng_deflate.c, 1242
- ng_deflate_config, 193
 - enable, 193
 - windowBits, 193
- NG_DEFLATE_CONFIG_INFO
 - ng_deflate.h, 1247
- ng_deflate_config_type
 - ng_deflate.c, 1244
- ng_deflate_config_type_fields
 - ng_deflate.c, 1244
- ng_deflate_constructor
 - ng_deflate.c, 1242, 1245
- ng_deflate_decompress
 - ng_deflate.c, 1242
- ng_deflate_disconnect
 - ng_deflate.c, 1243, 1245
- NG_DEFLATE_HOOK_COMP
 - ng_deflate.h, 1247
- NG_DEFLATE_HOOK_DECOMP
 - ng_deflate.h, 1247
- ng_deflate_newhook
 - ng_deflate.c, 1243, 1245
- NG_DEFLATE_NODE_TYPE
 - ng_deflate.h, 1248
- ng_deflate_private, 194
 - cfg, 194
 - compress, 194
 - ctrlnode, 194
 - cx, 194
 - inbuf, 194
 - outbuf, 194
 - seqnum, 195
 - stats, 195
- ng_deflate_rcvdata
 - ng_deflate.c, 1243, 1245
- ng_deflate_rcvmsg
 - ng_deflate.c, 1243, 1245
- ng_deflate_reset_req
 - ng_deflate.c, 1244
- ng_deflate_shutdown
 - ng_deflate.c, 1244, 1245
- ng_deflate_stat_type
 - ng_deflate.c, 1245
- ng_deflate_stats, 196
 - Errors, 196
 - FramesComp, 196
 - FramesPlain, 196
 - FramesUncomp, 196
 - InOctets, 196
 - OutOctets, 196
- NG_DEFLATE_STATS_INFO
 - ng_deflate.h, 1248
- ng_deflate_stats_type_fields
 - ng_deflate.c, 1245
- ng_deflate_tpestruct
 - ng_deflate.c, 1245
- ng_dequeue
 - ng_base.c, 1187
- ng_destroy_hook
 - ng_base.c, 1188
- ng_detach_common
 - ng_socket.c, 1584
- ng_device.c
 - DBG, 1251

- ERROUT, 1251
- MAX_NGD, 1251
- NETGRAPH_INIT, 1252
- ng_device_constructor, 1252, 1254
- ng_device_disconnect, 1252, 1254
- ng_device_mod_event, 1252
- ng_device_newhook, 1252, 1254
- ng_device_rcvdata, 1253, 1254
- ng_device_rcvmsg, 1253, 1254
- ng_device_shutdown, 1253, 1254
- ngd_cdevsw, 1254
- ngd_tpestruct, 1254
- ngd_unit, 1255
- ngdclose, 1253, 1255
- NGDF_OPEN, 1251
- NGDF_RWAIT, 1252
- ngdopen, 1253, 1255
- ngdpoll, 1253, 1255
- ngdread, 1253, 1255
- ngdwrite, 1253, 1255
- priv_p, 1252
- ng_device.h
 - NGM_DEVICE_GET_DEVNAME, 1256
- ng_device.h
 - NG_DEVICE_DEVNAME, 1256
 - NG_DEVICE_NODE_TYPE, 1256
 - NGM_DEVICE_COOKIE, 1256
- ng_device_constructor
 - ng_device.c, 1252, 1254
- NG_DEVICE_DEVNAME
 - ng_device.h, 1256
- ng_device_disconnect
 - ng_device.c, 1252, 1254
- ng_device_mod_event
 - ng_device.c, 1252
- ng_device_newhook
 - ng_device.c, 1252, 1254
- NG_DEVICE_NODE_TYPE
 - ng_device.h, 1256
- ng_device_rcvdata
 - ng_device.c, 1253, 1254
- ng_device_rcvmsg
 - ng_device.c, 1253, 1254
- ng_device_shutdown
 - ng_device.c, 1253, 1254
- ng_disconnect_t
 - netgraph.h, 1150
- ng_echo.c
 - NETGRAPH_INIT, 1258
 - nge_cons, 1258
 - nge_disconnect, 1258
 - nge_rcvdata, 1258
 - nge_rcvmsg, 1258
 - tpestruct, 1259
- ng_echo.h
 - NG_ECHO_NODE_TYPE, 1260
 - NGM_ECHO_COOKIE, 1260
- NG_ECHO_NODE_TYPE
 - ng_echo.h, 1260
- ng_eiface.c
 - NETGRAPH_INIT, 1263
 - ng_eiface_cmdlist, 1265
 - ng_eiface_constructor, 1263, 1266
 - ng_eiface_disconnect, 1264, 1266
 - ng_eiface_init, 1264
 - ng_eiface_ioctl, 1264
 - ng_eiface_mod_event, 1264
 - ng_eiface_newhook, 1264, 1266
 - ng_eiface_rcvdata, 1264, 1266
 - ng_eiface_rcvmsg, 1264, 1266
 - ng_eiface_rmnode, 1265, 1266
 - ng_eiface_start, 1265
 - ng_eiface_start2, 1265
 - ng_eiface_unit, 1266
 - priv_p, 1263
 - SA_SIZE, 1263
 - tpestruct, 1266
- ng_eiface.h
 - NGM{EIFACE_GET_IFADDRS, 1269
 - NGM{EIFACE_GET_IFNAME, 1269
 - NGM{EIFACE_SET, 1269
- ng_eiface.h
 - NG{EIFACE{EIFACE_NAME, 1268
 - NG{EIFACE{HOOK_ETHER, 1268
 - NG{EIFACE{MTU_DEFAULT, 1268
 - NG{EIFACE{MTU_MAX, 1268
 - NG{EIFACE{MTU_MIN, 1268
 - NG{EIFACE{NODE_TYPE, 1269
 - NGM{EIFACE{COOKIE, 1269
- ng_eiface_cmdlist
 - ng_eiface.c, 1265
- ng_eiface_constructor
 - ng_eiface.c, 1263, 1266
- ng_eiface_disconnect
 - ng_eiface.c, 1264, 1266
- NG{EIFACE{EIFACE_NAME
 - ng_eiface.h, 1268
- NG{EIFACE{HOOK_ETHER
 - ng_eiface.h, 1268
- ng_eiface_init
 - ng_eiface.c, 1264
- ng_eiface_ioctl
 - ng_eiface.c, 1264
- ng_eiface_mod_event
 - ng_eiface.c, 1264
- NG{EIFACE{MTU_DEFAULT
 - ng_eiface.h, 1268
- NG{EIFACE{MTU_MAX

- ng_eiface.h, 1268
- NG_EIFACE_MTU_MIN
 - ng_eiface.h, 1268
- ng_eiface_newhook
 - ng_eiface.c, 1264, 1266
- NG_EIFACE_NODE_TYPE
 - ng_eiface.h, 1269
- ng_eiface_private, 197
 - ether, 197
 - ifp, 197
 - node, 197
 - unit, 198
- ng_eiface_rcvdata
 - ng_eiface.c, 1264, 1266
- ng_eiface_rcvmsg
 - ng_eiface.c, 1264, 1266
- ng_eiface_rmnode
 - ng_eiface.c, 1265, 1266
- ng_eiface_start
 - ng_eiface.c, 1265
- ng_eiface_start2
 - ng_eiface.c, 1265
- ng_eiface_unit
 - ng_eiface.c, 1266
- ng_enaddr_parse
 - ng_parse.c, 1454
- ng_enaddr_unparse
 - ng_parse.c, 1455
- ng_encode_string
 - ng_parse.c, 1455
 - ng_parse.h, 1479
- ng_errseq
 - hva_stats_ng, 93
- ng_etf.c
 - etf_p, 1272
 - HASH, 1271
 - HASHSIZE, 1271
 - LIST_HEAD, 1272
 - M_NETGRAPH ETF, 1272
 - NETGRAPH_INIT, 1272
 - ng_etf_cmdlist, 1273
 - ng_etf_constructor, 1272, 1274
 - ng_etf_disconnect, 1272, 1274
 - ng_etf_filter_type, 1274
 - ng_etf_filter_type_fields, 1274
 - ng_etf_findentry, 1272
 - ng_etf_newhook, 1272, 1274
 - ng_etf_rcvdata, 1272, 1274
 - ng_etf_rcvmsg, 1273, 1274
 - ng_etf_shutdown, 1273, 1274
 - ng_etf_stat_type, 1275
 - ng_etf_stat_type_fields, 1275
 - typestruct, 1275
- ng_etf.h
 - NGM ETF_GET_STATUS, 1277
 - NGM ETF_SET_FILTER, 1277
 - NGM ETF_SET_FLAG, 1277
- ng_etf.h
 - NG ETF_FILTER_TYPE_INFO, 1276
 - NG ETF_HOOK_DOWNSTREAM, 1276
 - NG ETF_HOOK_NOMATCH, 1276
 - NG ETF_NODE_TYPE, 1276
 - NG ETF_STATS_TYPE_INFO, 1277
 - NGM ETF_COOKIE, 1277
- ng_etf_cmdlist
 - ng_etf.c, 1273
- ng_etf_constructor
 - ng_etf.c, 1272, 1274
- ng_etf_disconnect
 - ng_etf.c, 1272, 1274
- ng_etf_filter_type
 - ng_etf.c, 1274
- ng_etf_filter_type_fields
 - ng_etf.c, 1274
- NG ETF_FILTER_TYPE_INFO
 - ng_etf.h, 1276
- ng_etf_findentry
 - ng_etf.c, 1272
- NG ETF_HOOK_DOWNSTREAM
 - ng_etf.h, 1276
- NG ETF_HOOK_NOMATCH
 - ng_etf.h, 1276
- ng_etf_newhook
 - ng_etf.c, 1272, 1274
- NG ETF_NODE_TYPE
 - ng_etf.h, 1276
- ng_etf_rcvdata
 - ng_etf.c, 1272, 1274
- ng_etf_rcvmsg
 - ng_etf.c, 1273, 1274
- ng_etf_shutdown
 - ng_etf.c, 1273, 1274
- ng_etf_stat_type
 - ng_etf.c, 1275
- ng_etf_stat_type_fields
 - ng_etf.c, 1275
- NG ETF_STATS_TYPE_INFO
 - ng_etf.h, 1277
- ng_etffilter, 199
 - ethertype, 199
 - matchhook, 199
- ng_etfstat, 200
 - packets_in, 200
 - packets_out, 200
- ng_ether.c
 - IFP2NG, 1280
 - NETGRAPH_INIT, 1281
 - ng_ether_attach, 1281

- ng_ether_attach_p, 1284
- ng_ether_cmdlist, 1284
- ng_ether_constructor, 1281, 1284
- ng_ether_detach, 1281
- ng_ether_detach_p, 1284
- ng_ether_disconnect, 1281, 1284
- ng_ether_input, 1282
- ng_ether_input_orphan, 1282
- ng_ether_input_orphan_p, 1284
- ng_ether_input_p, 1284
- ng_ether_link_state, 1282
- ng_ether_link_state_p, 1284
- ng_ether_mod_event, 1282
- ng_ether_newhook, 1283, 1285
- ng_ether_output, 1283
- ng_ether_output_p, 1285
- ng_ether_rcv_lower, 1283
- ng_ether_rcv_upper, 1283
- ng_ether_rcvdata, 1283, 1285
- ng_ether_rcvmsg, 1283, 1285
- ng_ether_shutdown, 1284, 1285
- ng_ether_ttypestruct, 1285
- priv_p, 1280
- ng_ether.h
 - NGM_ETHER_ADD_MULTI, 1287
 - NGM_ETHER_DEL_MULTI, 1287
 - NGM_ETHER_DETACH, 1287
 - NGM_ETHER_GET_AUTOSRC, 1287
 - NGM_ETHER_GET_ENADDR, 1287
 - NGM_ETHER_GET_IFINDEX, 1287
 - NGM_ETHER_GET_IFNAME, 1287
 - NGM_ETHER_GET_PROMISC, 1287
 - NGM_ETHER_SET_AUTOSRC, 1287
 - NGM_ETHER_SET_ENADDR, 1287
 - NGM_ETHER_SET_PROMISC, 1287
- ng_ether.h
 - NG_ETHER_HOOK_DIVERT, 1286
 - NG_ETHER_HOOK_LOWER, 1286
 - NG_ETHER_HOOK_ORPHAN, 1286
 - NG_ETHER_HOOK_UPPER, 1286
 - NG_ETHER_NODE_TYPE, 1287
 - NGM_ETHER_COOKIE, 1287
- ng_ether_attach
 - ng_ether.c, 1281
- ng_ether_attach_p
 - ng_ether.c, 1284
- ng_ether_cmdlist
 - ng_ether.c, 1284
- ng_ether_constructor
 - ng_ether.c, 1281, 1284
- ng_ether_detach
 - ng_ether.c, 1281
- ng_ether_detach_p
 - ng_ether.c, 1284
- ng_ether_disconnect
 - ng_ether.c, 1281, 1284
- NG_ETHER_HOOK_DIVERT
 - ng_ether.h, 1286
- NG_ETHER_HOOK_LOWER
 - ng_ether.h, 1286
- NG_ETHER_HOOK_ORPHAN
 - ng_ether.h, 1286
- NG_ETHER_HOOK_UPPER
 - ng_ether.h, 1286
- ng_ether_input
 - ng_ether.c, 1282
- ng_ether_input_orphan
 - ng_ether.c, 1282
- ng_ether_input_orphan_p
 - ng_ether.c, 1284
- ng_ether_input_p
 - ng_ether.c, 1284
- ng_ether_link_state
 - ng_ether.c, 1282
- ng_ether_link_state_p
 - ng_ether.c, 1284
- ng_ether_mod_event
 - ng_ether.c, 1282
- ng_ether_newhook
 - ng_ether.c, 1283, 1285
- NG_ETHER_NODE_TYPE
 - ng_ether.h, 1287
- ng_ether_output
 - ng_ether.c, 1283
- ng_ether_output_p
 - ng_ether.c, 1285
- ng_ether_rcv_lower
 - ng_ether.c, 1283
- ng_ether_rcv_upper
 - ng_ether.c, 1283
- ng_ether_rcvdata
 - ng_ether.c, 1283, 1285
- ng_ether_rcvmsg
 - ng_ether.c, 1283, 1285
- ng_ether_shutdown
 - ng_ether.c, 1284, 1285
- ng_ether_ttypestruct
 - ng_ether.c, 1285
- ng_fec.c
 - FEC_BTYPE_INET, 1290
 - FEC_BTYPE_INET6, 1290
 - FEC_BTYPE_MAC, 1291
 - FEC_BUNDLESIZ, 1291
 - IFP2NG, 1291
 - NETGRAPH_INIT, 1291
 - ng_fec_addport, 1291
 - ng_fec_choose_port, 1291
 - ng_fec_cmds, 1295

- ng_fec_constructor, 1292, 1295
- ng_fec_delpport, 1292
- ng_fec_free_unit, 1292
- ng_fec_get_unit, 1292
- ng_fec_ifmedia_sts, 1293
- ng_fec_ifmedia_upd, 1293
- ng_fec_init, 1293
- ng_fec_input, 1293
- ng_fec_ioctl, 1293
- ng_fec_mod_event, 1294
- ng_fec_mtx, 1296
- ng_fec_output, 1294
- ng_fec_rcvmsg, 1294, 1296
- ng_fec_setport, 1294
- ng_fec_shutdown, 1294, 1296
- ng_fec_start, 1295
- ng_fec_stop, 1295
- ng_fec_tick, 1295
- ng_fec_units, 1296
- ng_fec_units_len, 1296
- ng_units_in_use, 1296
- priv_p, 1291
- typestruct, 1296
- UNITS_BITSPERWORD, 1291
- ng_fec.h
 - NGM_FEC_ADD_IFACE, 1299
 - NGM_FEC_DEL_IFACE, 1299
 - NGM_FEC_SET_MODE_INET, 1299
 - NGM_FEC_SET_MODE_INET6, 1299
 - NGM_FEC_SET_MODE_MAC, 1299
- ng_fec.h
 - M_FEC_INET, 1297
 - M_FEC_INET6, 1297
 - M_FEC_MAC, 1297
 - NETISR_FEC, 1298
 - NG_FEC_FEC_NAME, 1298
 - NG_FEC_FEC_NAME_MAX, 1298
 - NG_FEC_MTU_DEFAULT, 1298
 - NG_FEC_MTU_MAX, 1298
 - NG_FEC_MTU_MIN, 1298
 - NG_FEC_NODE_TYPE, 1298
 - NGM_FEC_COOKIE, 1298
- ng_fec_addport
 - ng_fec.c, 1291
- ng_fec_bundle, 201
- ng_fec_choose_port
 - ng_fec.c, 1291
- ng_fec_cmds
 - ng_fec.c, 1295
- ng_fec_constructor
 - ng_fec.c, 1292, 1295
- ng_fec_delpport
 - ng_fec.c, 1292
- NG_FEC_FEC_NAME
 - ng_fec.h, 1298
- NG_FEC_FEC_NAME_MAX
 - ng_fec.h, 1298
- ng_fec_free_unit
 - ng_fec.c, 1292
- ng_fec_get_unit
 - ng_fec.c, 1292
- ng_fec_ifmedia_sts
 - ng_fec.c, 1293
- ng_fec_ifmedia_upd
 - ng_fec.c, 1293
- ng_fec_ifname, 202
 - ngif_name, 202
- ng_fec_init
 - ng_fec.c, 1293
- ng_fec_input
 - ng_fec.c, 1293
- ng_fec_ioctl
 - ng_fec.c, 1293
- ng_fec_mod_event
 - ng_fec.c, 1294
- NG_FEC_MTU_DEFAULT
 - ng_fec.h, 1298
- NG_FEC_MTU_MAX
 - ng_fec.h, 1298
- NG_FEC_MTU_MIN
 - ng_fec.h, 1298
- ng_fec_mtx
 - ng_fec.c, 1296
- NG_FEC_NODE_TYPE
 - ng_fec.h, 1298
- ng_fec_output
 - ng_fec.c, 1294
- ng_fec_portlist, 203
 - fec_idx, 203
 - fec_if, 203
 - fec_if_input, 203
 - fec_ifstat, 203
 - fec_mac, 203
- ng_fec_private, 204
 - fec_bundle, 204
 - fec_ch, 204
 - if_error, 205
 - if_flags, 205
 - ifmedia, 205
 - ifp, 205
 - node, 205
 - unit, 205
- ng_fec_rcvmsg
 - ng_fec.c, 1294, 1296
- ng_fec_setport
 - ng_fec.c, 1294
- ng_fec_shutdown
 - ng_fec.c, 1294, 1296

- ng_fec_start
 - ng_fec.c, 1295
- ng_fec_stop
 - ng_fec.c, 1295
- ng_fec_tick
 - ng_fec.c, 1295
- ng_fec_units
 - ng_fec.c, 1296
- ng_fec_units_len
 - ng_fec.c, 1296
- ng_findhook
 - netgraph.h, 1152
 - ng_base.c, 1188
- ng_findhook_t
 - netgraph.h, 1150
- ng_findtype
 - netgraph.h, 1152
 - ng_base.c, 1188
- ng_fixedarray_getAlign
 - ng_parse.c, 1455
- ng_fixedarray_getDefault
 - ng_parse.c, 1455
- ng_fixedarray_parse
 - ng_parse.c, 1455
- ng_fixedarray_unparse
 - ng_parse.c, 1456
- ng_fixedstring_getDefault
 - ng_parse.c, 1456
- ng_fixedstring_parse
 - ng_parse.c, 1456
- ng_fixedstring_unparse
 - ng_parse.c, 1456
- ng_flush_input_queue
 - ng_base.c, 1188
- ng_fn_eachhook
 - netgraph.h, 1150
- NG_FORCE_WRITER
 - netgraph.h, 1128
- ng_frame_relay.c
 - BYTE1_C_R, 1301
 - BYTE2_BECD, 1301
 - BYTE2_DE, 1301
 - BYTE2_FECN, 1302
 - BYTEX_EA, 1302
 - CHAN_ACTIVE, 1302
 - CHAN_VALID, 1302
 - CTX_VALID, 1302
 - CTX_VALUE, 1302
 - LASTBYTE_D_C, 1302
 - makeup, 1305
 - MAX_CT, 1302
 - NETGRAPH_INIT, 1303
 - ngfrm_addrln, 1303
 - ngfrm_allocate_CTX, 1303
 - ngfrm_constructor, 1303, 1305
 - ngfrm_decode, 1304
 - ngfrm_disconnect, 1304, 1305
 - ngfrm_newhook, 1304, 1305
 - ngfrm_rcvdata, 1304, 1305
 - ngfrm_shutdown, 1305
 - sc_p, 1303
 - SHIFTIN, 1302
 - SHIFTOUT, 1303
 - typestruct, 1305
- ng_frame_relay.h
 - NG_FRAMERELAY_HOOK_DEBUG, 1307
 - NG_FRAMERELAY_HOOK_DLCI, 1307
 - NG_FRAMERELAY_HOOK_-,
DOWNSTREAM, 1307
 - NG_FRAMERELAY_NODE_TYPE, 1307
 - NGM_FRAMERELAY_COOKIE, 1307
- NG_FRAMERELAY_HOOK_DEBUG
 - ng_frame_relay.h, 1307
- NG_FRAMERELAY_HOOK_DLCI
 - ng_frame_relay.h, 1307
- NG_FRAMERELAY_HOOK_DOWNSTREAM
 - ng_frame_relay.h, 1307
- NG_FRAMERELAY_NODE_TYPE
 - ng_frame_relay.h, 1307
- NG_FREE_HOOK
 - ng_base.c, 1180
- NG_FREE_ITEM
 - netgraph.h, 1128
- ng_free_item
 - netgraph.h, 1152
 - ng_base.c, 1189
- NG_FREE_M
 - netgraph.h, 1129
- NG_FREE_META
 - netgraph.h, 1129
- NG_FREE_MSG
 - netgraph.h, 1130
- NG_FREE_NODE
 - ng_base.c, 1180
- NG_FWD_ITEM_HOOK
 - netgraph.h, 1130
- NG_FWD_ITEM_HOOK_FLAGS
 - netgraph.h, 1130
- NG_FWD_MSG_HOOK
 - netgraph.h, 1131
- NG_FWD_NEW_DATA
 - netgraph.h, 1131
- NG_FWD_NEW_DATA_FLAGS
 - netgraph.h, 1131
- NG_GENERIC_BANDWIDTH_INFO
 - ng_message.h, 1410
- ng_generic_cmds
 - ng_base.c, 1202

- NG_GENERIC_CONNECT_INFO
 - ng_message.h, 1410
- NG_GENERIC_FLOW_MANAGER_INFO
 - ng_message.h, 1410
- NG_GENERIC_HOOKLIST_INFO
 - ng_message.h, 1411
- ng_generic_linkinfo_array_type
 - ng_base.c, 1202
- ng_generic_linkinfo_array_type_info
 - ng_base.c, 1202
- ng_generic_linkinfo_getLength
 - ng_base.c, 1189
- NG_GENERIC_LINKINFO_INFO
 - ng_message.h, 1411
- NG_GENERIC_LISTNODES_INFO
 - ng_message.h, 1411
- NG_GENERIC_MKPEER_INFO
 - ng_message.h, 1411
- ng_generic_msg
 - ng_base.c, 1189
- NG_GENERIC_NAME_INFO
 - ng_message.h, 1411
- NG_GENERIC_NG_MSG_INFO
 - ng_message.h, 1412
- NG_GENERIC_NODEINFO_INFO
 - ng_message.h, 1412
- ng_generic_nodeinfoarray_type
 - ng_base.c, 1202
- NG_GENERIC_QUEUE_INFO
 - ng_message.h, 1412
- NG_GENERIC_RMHOOK_INFO
 - ng_message.h, 1412
- NG_GENERIC_TYPEINFO_INFO
 - ng_message.h, 1413
- ng_generic_typeinfoarray_type
 - ng_base.c, 1202
- NG_GENERIC_TPELIST_INFO
 - ng_message.h, 1413
- ng_get_composite_elem_default
 - ng_parse.c, 1457
- ng_get_composite_etype
 - ng_parse.c, 1457
- ng_get_composite_len
 - ng_parse.c, 1457
- ng_get_getAlign_method
 - ng_parse.c, 1457
- ng_get_getDefault_method
 - ng_parse.c, 1458
- ng_get_parse_method
 - ng_parse.c, 1458
- ng_get_string_token
 - ng_parse.c, 1458
 - ng_parse.h, 1479
- ng_get_unparse_method
 - ng_parse.c, 1458
- ng_getAlign_t
 - ng_parse.h, 1478
- ng_getDefault_t
 - ng_parse.h, 1478
- ng_getqblk
 - ng_base.c, 1190
- ng_gif.c
 - IFP2NG, 1310
 - IFP2NG_SET, 1310
 - MODULE_DEPEND, 1311
 - NETGRAPH_INIT, 1311
 - ng_gif_attach, 1311
 - ng_gif_cmdlist, 1314
 - ng_gif_connect, 1311, 1314
 - ng_gif_constructor, 1311, 1314
 - ng_gif_detach, 1311
 - ng_gif_disconnect, 1311, 1314
 - ng_gif_glue_af, 1312
 - ng_gif_input, 1312
 - ng_gif_input2, 1312
 - ng_gif_input_orphan, 1312
 - ng_gif_mod_event, 1313
 - ng_gif_newhook, 1313, 1314
 - ng_gif_rcv_lower, 1313
 - ng_gif_rcvdata, 1313, 1315
 - ng_gif_rcvmsg, 1314, 1315
 - ng_gif_shutdown, 1314, 1315
 - ng_gif_typestruct, 1315
 - priv_p, 1310
- ng_gif.h
 - NGM_GIF_GET_IFINDEX, 1317
 - NGM_GIF_GET_IFNAME, 1317
- ng_gif.h
 - NG_GIF_HOOK_DIVERT, 1316
 - NG_GIF_HOOK_LOWER, 1316
 - NG_GIF_HOOK_ORPHAN, 1316
 - NG_GIF_NODE_TYPE, 1316
 - NGM_GIF_COOKIE, 1316
- ng_gif_attach
 - ng_gif.c, 1311
- ng_gif_cmdlist
 - ng_gif.c, 1314
- ng_gif_connect
 - ng_gif.c, 1311, 1314
- ng_gif_constructor
 - ng_gif.c, 1311, 1314
- ng_gif_demux.c
 - get_hook_from_iffam, 1320
 - get_iffam_from_af, 1320
 - get_iffam_from_hook, 1320
 - get_iffam_from_name, 1320
 - gFamilies, 1322
 - iffam_p, 1320

- M_NETGRAPH_GIF_DEMUX, 1319
- NETGRAPH_INIT, 1320
- ng_gif_demux_cmdlist, 1322
- ng_gif_demux_constructor, 1320, 1322
- ng_gif_demux_disconnect, 1320, 1322
- ng_gif_demux_newhook, 1321, 1322
- ng_gif_demux_rcvdata, 1321, 1322
- ng_gif_demux_rcvmsg, 1321, 1322
- ng_gif_demux_shutdown, 1321, 1323
- ng_gif_demux_tpestruct, 1323
- NUM_FAMILIES, 1319
- priv_p, 1320
- ng_gif_demux.h
 - NG_GIF_DEMUX_HOOK_ATALK, 1324
 - NG_GIF_DEMUX_HOOK_ATM, 1324
 - NG_GIF_DEMUX_HOOK_GIF, 1324
 - NG_GIF_DEMUX_HOOK_INET, 1324
 - NG_GIF_DEMUX_HOOK_INET6, 1324
 - NG_GIF_DEMUX_HOOK_IPX, 1324
 - NG_GIF_DEMUX_HOOK_NATM, 1324
 - NG_GIF_DEMUX_NODE_TYPE, 1325
 - NGM_GIF_DEMUX_COOKIE, 1325
- ng_gif_demux_cmdlist
 - ng_gif_demux.c, 1322
- ng_gif_demux_constructor
 - ng_gif_demux.c, 1320, 1322
- ng_gif_demux_disconnect
 - ng_gif_demux.c, 1320, 1322
- NG_GIF_DEMUX_HOOK_ATALK
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_ATM
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_GIF
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_INET
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_INET6
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_IPX
 - ng_gif_demux.h, 1324
- NG_GIF_DEMUX_HOOK_NATM
 - ng_gif_demux.h, 1324
- ng_gif_demux_newhook
 - ng_gif_demux.c, 1321, 1322
- NG_GIF_DEMUX_NODE_TYPE
 - ng_gif_demux.h, 1325
- ng_gif_demux_private, 206
 - gif, 206
 - hooks, 206
 - node, 206
- ng_gif_demux_rcvdata
 - ng_gif_demux.c, 1321, 1322
- ng_gif_demux_rcvmsg
 - ng_gif_demux.c, 1321, 1322
- ng_gif_demux_shutdown
 - ng_gif_demux.c, 1321, 1323
- ng_gif_demux_tpestruct
 - ng_gif_demux.c, 1323
- ng_gif_detach
 - ng_gif.c, 1311
- ng_gif_disconnect
 - ng_gif.c, 1311, 1314
- ng_gif_glue_af
 - ng_gif.c, 1312
- NG_GIF_HOOK_DIVERT
 - ng_gif.h, 1316
- NG_GIF_HOOK_LOWER
 - ng_gif.h, 1316
- NG_GIF_HOOK_ORPHAN
 - ng_gif.h, 1316
- ng_gif_input
 - ng_gif.c, 1312
- ng_gif_input2
 - ng_gif.c, 1312
- ng_gif_input_orphan
 - ng_gif.c, 1312
- ng_gif_mod_event
 - ng_gif.c, 1313
- ng_gif_newhook
 - ng_gif.c, 1313, 1314
- NG_GIF_NODE_TYPE
 - ng_gif.h, 1316
- ng_gif_rcv_lower
 - ng_gif.c, 1313
- ng_gif_rcvdata
 - ng_gif.c, 1313, 1315
- ng_gif_rcvmsg
 - ng_gif.c, 1314, 1315
- ng_gif_shutdown
 - ng_gif.c, 1314, 1315
- ng_gif_tpestruct
 - ng_gif.c, 1315
- ng_h4.c
 - MALLOC_DEFINE, 680
 - MODULE_VERSION, 680
 - NET_NEEDS_GIANT, 680
 - NETGRAPH_INIT, 680
 - ng_h4_close, 680
 - ng_h4_connect, 681, 685
 - ng_h4_constructor, 681, 685
 - ng_h4_disc, 685
 - ng_h4_disconnect, 681, 685
 - ng_h4_input, 681
 - ng_h4_ioctl, 681
 - ng_h4_mod_event, 682
 - ng_h4_newhook, 682, 685
 - ng_h4_node, 685
 - ng_h4_open, 682

- ng_h4_process_timeout, 682
- ng_h4_rcvdata, 682, 686
- ng_h4_rcvmsg, 683, 686
- ng_h4_read, 683
- ng_h4_shutdown, 683, 686
- ng_h4_start, 683
- ng_h4_start2, 684
- ng_h4_timeout, 684
- ng_h4_untimeout, 684
- ng_h4_write, 685
- NI, 680
- typestruct, 686
- ng_h4.h
 - NG_H4_ALERT_LEVEL, 843
 - NG_H4_ERR_LEVEL, 843
 - NG_H4_HOOK, 843
 - NG_H4_INFO_LEVEL, 844
 - ng_h4_node_debug_ep, 845
 - ng_h4_node_qlen_ep, 845
 - ng_h4_node_state_ep, 845
 - NG_H4_NODE_TYPE, 844
 - NG_H4_W4_PKT_DATA, 844
 - NG_H4_W4_PKT_HDR, 844
 - NG_H4_W4_PKT_IND, 844
 - NG_H4_WARN_LEVEL, 844
 - NGM_H4_COOKIE, 844
 - NGM_H4_NODE_GET_DEBUG, 844
 - NGM_H4_NODE_GET_QLEN, 844
 - NGM_H4_NODE_GET_STAT, 845
 - NGM_H4_NODE_GET_STATE, 845
 - NGM_H4_NODE_RESET, 845
 - NGM_H4_NODE_RESET_STAT, 845
 - NGM_H4_NODE_SET_DEBUG, 845
 - NGM_H4_NODE_SET_QLEN, 845
- NG_H4_ALERT
 - ng_h4_var.h, 688
- NG_H4_ALERT_LEVEL
 - ng_h4.h, 843
- ng_h4_close
 - ng_h4.c, 680
- ng_h4_cmdlist
 - ng_h4_prse.h, 687
- ng_h4_connect
 - ng_h4.c, 681, 685
- ng_h4_constructor
 - ng_h4.c, 681, 685
- NG_H4_DEFAULTQLEN
 - ng_h4_var.h, 688
- ng_h4_disc
 - ng_h4.c, 685
- ng_h4_disconnect
 - ng_h4.c, 681, 685
- NG_H4_ERR
 - ng_h4_var.h, 689
- NG_H4_ERR_LEVEL
 - ng_h4.h, 843
- NG_H4_HIWATER
 - ng_h4_var.h, 689
- NG_H4_HOOK
 - ng_h4.h, 843
- NG_H4_IBUF_SIZE
 - ng_h4_var.h, 689
- NG_H4_INFO
 - ng_h4_var.h, 689
- ng_h4_info, 208
 - debug, 209
 - flags, 209
 - got, 209
 - hook, 209
 - ibuf, 209
 - node, 209
 - outq, 209
 - stat, 209
 - state, 209
 - timo, 210
 - tp, 210
 - want, 210
- NG_H4_INFO_LEVEL
 - ng_h4.h, 844
- ng_h4_info_p
 - ng_h4_var.h, 690
- ng_h4_info_t
 - ng_h4_var.h, 690
- ng_h4_input
 - ng_h4.c, 681
- ng_h4_ioctl
 - ng_h4.c, 681
- ng_h4_mod_event
 - ng_h4.c, 682
- ng_h4_newhook
 - ng_h4.c, 682, 685
- ng_h4_node
 - ng_h4.c, 685
- ng_h4_node_debug_ep
 - ng_h4.h, 845
- ng_h4_node_qlen_ep
 - ng_h4.h, 845
- ng_h4_node_stat_ep, 211
 - bytes_recv, 211
 - bytes_sent, 211
 - ierrors, 211
 - oerrors, 211
 - pckts_recv, 211
 - pckts_sent, 211
- ng_h4_node_state_ep
 - ng_h4.h, 845
- NG_H4_NODE_TYPE
 - ng_h4.h, 844

- ng_h4_open
 - ng_h4.c, 682
- ng_h4_process_timeout
 - ng_h4.c, 682
- ng_h4_prse.h
 - ng_h4_cmdlist, 687
 - ng_h4_stat_type, 687
 - ng_h4_stat_type_fields, 687
- ng_h4_rcvdata
 - ng_h4.c, 682, 686
- ng_h4_rcvmsg
 - ng_h4.c, 683, 686
- ng_h4_read
 - ng_h4.c, 683
- ng_h4_shutdown
 - ng_h4.c, 683, 686
- ng_h4_start
 - ng_h4.c, 683
- ng_h4_start2
 - ng_h4.c, 684
- NG_H4_STAT_BYTES_RECV
 - ng_h4_var.h, 689
- NG_H4_STAT_BYTES_SENT
 - ng_h4_var.h, 689
- NG_H4_STAT_IERROR
 - ng_h4_var.h, 689
- NG_H4_STAT_OERROR
 - ng_h4_var.h, 689
- NG_H4_STAT_PCKTS_RECV
 - ng_h4_var.h, 689
- NG_H4_STAT_PCKTS_SENT
 - ng_h4_var.h, 690
- NG_H4_STAT_RESET
 - ng_h4_var.h, 690
- ng_h4_stat_type
 - ng_h4_prse.h, 687
- ng_h4_stat_type_fields
 - ng_h4_prse.h, 687
- NG_H4_TIMEOUT
 - ng_h4_var.h, 690
- ng_h4_timeout
 - ng_h4.c, 684
- ng_h4_untimeout
 - ng_h4.c, 684
- ng_h4_var.h
 - MALLOC_DECLARE, 690
 - NG_H4_ALERT, 688
 - NG_H4_DEFAULTQLEN, 688
 - NG_H4_ERR, 689
 - NG_H4_HIWATER, 689
 - NG_H4_IBUF_SIZE, 689
 - NG_H4_INFO, 689
 - ng_h4_info_p, 690
 - ng_h4_info_t, 690
 - NG_H4_STAT_BYTES_RECV, 689
 - NG_H4_STAT_BYTES_SENT, 689
 - NG_H4_STAT_IERROR, 689
 - NG_H4_STAT_OERROR, 689
 - NG_H4_STAT_PCKTS_RECV, 689
 - NG_H4_STAT_PCKTS_SENT, 690
 - NG_H4_STAT_RESET, 690
 - NG_H4_TIMEOUT, 690
 - NG_H4_WARN, 690
- NG_H4_W4_PKT_DATA
 - ng_h4.h, 844
- NG_H4_W4_PKT_HDR
 - ng_h4.h, 844
- NG_H4_W4_PKT_IND
 - ng_h4.h, 844
- NG_H4_WARN
 - ng_h4_var.h, 690
- NG_H4_WARN_LEVEL
 - ng_h4.h, 844
- ng_h4_write
 - ng_h4.c, 685
- ng_hci.h
 - bdaddr_p, 896
 - NG_HCI_ACL_DATA_PKT, 859
 - NG_HCI_ACL_PKT_SIZE, 859
 - NG_HCI_ALERT_LEVEL, 859
 - NG_HCI_BC_FLAG, 859
 - NG_HCI_BDADDR_ANY, 859
 - NG_HCI_BDADDR_SIZE, 859
 - NG_HCI_BROADCAST_ACTIVE, 859
 - NG_HCI_BROADCAST_PICONET, 859
 - ng_hci_change_local_name_rp, 896
 - NG_HCI_CLASS_SIZE, 859
 - NG_HCI_CMD_PKT, 859
 - NG_HCI_CMD_PKT_SIZE, 860
 - NG_HCI_CON_CLOSED, 860
 - NG_HCI_CON_HANDLE, 860
 - NG_HCI_CON_OPEN, 860
 - NG_HCI_CON_W4_CONN_COMPLETE, 860
 - NG_HCI_CON_W4_LP_CON_RSP, 860
 - NG_HCI_COUNTRY_CODE_FRANCE, 860
 - NG_HCI_COUNTRY_CODE_NAM_EUR_JP, 860
 - ng_hci_create_new_unit_key_rp, 896
 - ng_hci_enable_unit_under_test_rp, 897
 - NG_HCI_ENCRYPTION_MODE_ALL, 861
 - NG_HCI_ENCRYPTION_MODE_NONE, 861
 - NG_HCI_ENCRYPTION_MODE_P2P, 861
 - NG_HCI_ERR_LEVEL, 861
 - NG_HCI_EVENT_AUTH_COMPL, 861
 - NG_HCI_EVENT_BT_LOGO, 861

- NG_HCI_EVENT_CHANGE_CON_LINK_-
KEY_COMPL, 861
- NG_HCI_EVENT_COMMAND_COMPL,
861
- NG_HCI_EVENT_COMMAND_STATUS,
861
- NG_HCI_EVENT_CON_COMPL, 862
- NG_HCI_EVENT_CON_PKT_TYPE_-
CHANGED, 862
- NG_HCI_EVENT_CON_REQ, 862
- NG_HCI_EVENT_DATA_BUFFER_-
OVERFLOW, 862
- NG_HCI_EVENT_DISCON_COMPL, 862
- NG_HCI_EVENT_ENCRYPTION_-
CHANGE, 862
- NG_HCI_EVENT_FLUSH_OCCUR, 862
- NG_HCI_EVENT_HARDWARE_ERROR,
862
- NG_HCI_EVENT_INQUIRY_COMPL, 862
- NG_HCI_EVENT_INQUIRY_RESULT, 863
- NG_HCI_EVENT_LINK_KEY_-
NOTIFICATION, 863
- NG_HCI_EVENT_LINK_KEY_REQ, 863
- NG_HCI_EVENT_LOOPBACK_-
COMMAND, 863
- NG_HCI_EVENT_MASK_SIZE, 863
- NG_HCI_EVENT_MASTER_LINK_KEY_-
COMPL, 863
- NG_HCI_EVENT_MAX_SLOT_CHANGE,
863
- NG_HCI_EVENT_MODE_CHANGE, 863
- NG_HCI_EVENT_NUM_COMPL_PKTS,
863
- NG_HCI_EVENT_PAGE_SCAN_MODE_-
CHANGE, 864
- NG_HCI_EVENT_PAGE_SCAN_REP_-
MODE_CHANGE, 864
- NG_HCI_EVENT_PIN_CODE_REQ, 864
- NG_HCI_EVENT_PKT, 864
- NG_HCI_EVENT_PKT_SIZE, 864
- NG_HCI_EVENT_QOS_SETUP_COMPL,
864
- NG_HCI_EVENT_QOS_VIOLATION, 864
- NG_HCI_EVENT_READ_CLOCK_-
OFFSET_COMPL, 864
- NG_HCI_EVENT_READ_REMOTE_-
FEATURES_COMPL, 864
- NG_HCI_EVENT_READ_REMOTE_VER_-
INFO_COMPL, 865
- NG_HCI_EVENT_REMOTE_NAME_-
REQ_COMPL, 865
- NG_HCI_EVENT_RETURN_LINK_KEYS,
865
- NG_HCI_EVENT_ROLE_CHANGE, 865
- NG_HCI_EVENT_VENDOR, 865
- NG_HCI_EVMSK_ALL, 865
- NG_HCI_EVMSK_AUTH_COMPL, 865
- NG_HCI_EVMSK_CHANGE_CON_LINK_-
KEY_COMPL, 865
- NG_HCI_EVMSK_COMMAND_COMPL,
865
- NG_HCI_EVMSK_COMMAND_STATUS,
866
- NG_HCI_EVMSK_CON_COMPL, 866
- NG_HCI_EVMSK_CON_PKT_TYPE_-
CHANGED, 866
- NG_HCI_EVMSK_CON_REQ, 866
- NG_HCI_EVMSK_DATA_BUFFER_-
OVERFLOW, 866
- NG_HCI_EVMSK_DISCON_COMPL, 866
- NG_HCI_EVMSK_ENCRYPTION_-
CHANGE, 866
- NG_HCI_EVMSK_FLUSH_OCCUR, 866
- NG_HCI_EVMSK_HARDWARE_ERROR,
866
- NG_HCI_EVMSK_INQUIRY_COMPL, 866
- NG_HCI_EVMSK_INQUIRY_RESULT, 866
- NG_HCI_EVMSK_LINK_KEY_-
NOTIFICATION, 867
- NG_HCI_EVMSK_LINK_KEY_REQ, 867
- NG_HCI_EVMSK_LOOPBACK_-
COMMAND, 867
- NG_HCI_EVMSK_MASTER_LINK_KEY_-
COMPL, 867
- NG_HCI_EVMSK_MAX_SLOT_CHANGE,
867
- NG_HCI_EVMSK_MODE_CHANGE, 867
- NG_HCI_EVMSK_NONE, 867
- NG_HCI_EVMSK_NUM_COMPL_PKTS,
867
- NG_HCI_EVMSK_PAGE_SCAN_MODE_-
CHANGE, 867
- NG_HCI_EVMSK_PAGE_SCAN_REP_-
MODE_CHANGE, 867
- NG_HCI_EVMSK_PIN_CODE_REQ, 867
- NG_HCI_EVMSK_QOS_SETUP_COMPL,
868
- NG_HCI_EVMSK_QOS_VIOLATION, 868
- NG_HCI_EVMSK_READ_CLOCK_-
OFFSET_COMLETE, 868
- NG_HCI_EVMSK_READ_REMOTE_-
FEATURES_COMPL, 868
- NG_HCI_EVMSK_READ_REMOTE_-
VER_INFO_COMPL, 868
- NG_HCI_EVMSK_REMOTE_NAME_-
REQ_COMPL, 868
- NG_HCI_EVMSK_RETURN_LINK_KEYS,
868

- NG_HCI_EVMSK_ROLE_CHANGE, 868
- ng_hci_exit_periodic_inquiry_rp, 897
- NG_HCI_FEATURES_SIZE, 868
- NG_HCI_FILTER_COND_CON_ANY_-
UNIT, 868
- NG_HCI_FILTER_COND_CON_BDADDR,
868
- NG_HCI_FILTER_COND_CON_UNIT_-
CLASS, 869
- NG_HCI_FILTER_COND_INQUIRY_-
BDADDR, 869
- NG_HCI_FILTER_COND_INQUIRY_-
NEW_UNIT, 869
- NG_HCI_FILTER_COND_INQUIRY_-
UNIT_CLASS, 869
- NG_HCI_FILTER_TYPE_CON_SETUP, 869
- NG_HCI_FILTER_TYPE_INQUIRY_-
RESULT, 869
- NG_HCI_FILTER_TYPE_NONE, 869
- NG_HCI_H2HC_FLOW_CONTROL_ACL,
869
- NG_HCI_H2HC_FLOW_CONTROL_BOTH,
869
- NG_HCI_H2HC_FLOW_CONTROL_-
NONE, 869
- ng_hci_h2hc_flow_control_rp, 897
- NG_HCI_H2HC_FLOW_CONTROL_SCO,
869
- NG_HCI_HOLD_MODE_NO_CHANGE,
870
- NG_HCI_HOLD_MODE_SUSPEND_-
INQUIRY_SCAN, 870
- NG_HCI_HOLD_MODE_SUSPEND_-
PAGE_SCAN, 870
- NG_HCI_HOLD_MODE_SUSPEND_-
PERIOD_INQUIRY, 870
- NG_HCI_HOOK_ACL, 870
- NG_HCI_HOOK_DRV, 870
- NG_HCI_HOOK_RAW, 870
- NG_HCI_HOOK_SCO, 870
- ng_hci_host_buffer_size_rp, 897
- NG_HCI_INFO_LEVEL, 870
- ng_hci_inquiry_cancel_rp, 897
- NG_HCI_INQUIRY_DISABLE_PAGE_-
ENABLE, 870
- NG_HCI_INQUIRY_ENABLE_PAGE_-
DISABLE, 871
- NG_HCI_INQUIRY_ENABLE_PAGE_-
ENABLE, 871
- NG_HCI_KEY_SIZE, 871
- NG_HCI_LAP_SIZE, 871
- NG_HCI_LINK_ACL, 871
- NG_HCI_LINK_KEY_TYPE_-
COMBINATION_KEY, 871
- NG_HCI_LINK_KEY_TYPE_LOCAL_-
UNIT_KEY, 871
- NG_HCI_LINK_KEY_TYPE_REMOTE_-
UNIT_KEY, 871
- NG_HCI_LINK_POLICY_DISABLE_ALL_-
LM_MODES, 871
- NG_HCI_LINK_POLICY_ENABLE_-
HOLD_MODE, 871
- NG_HCI_LINK_POLICY_ENABLE_-
PARK_MODE, 872
- NG_HCI_LINK_POLICY_ENABLE_-
ROLE_SWITCH, 872
- NG_HCI_LINK_POLICY_ENABLE_-
SNIFF_MODE, 872
- NG_HCI_LINK_SCO, 872
- NG_HCI_LMP_3SLOT, 872
- NG_HCI_LMP_5SLOT, 872
- NG_HCI_LMP_ALAW_LOG, 872
- NG_HCI_LMP_CHANNEL_QUALITY, 872
- NG_HCI_LMP_CVSD, 872
- NG_HCI_LMP_ENCRYPTION, 872
- NG_HCI_LMP_FLOW_CONTROL_LAG0,
873
- NG_HCI_LMP_FLOW_CONTROL_LAG1,
873
- NG_HCI_LMP_FLOW_CONTROL_LAG2,
873
- NG_HCI_LMP_HOLD_MODE, 873
- NG_HCI_LMP_HV2_PKT, 873
- NG_HCI_LMP_HV3_PKT, 873
- NG_HCI_LMP_PAGING_SCHEME, 873
- NG_HCI_LMP_PARK_MODE, 873
- NG_HCI_LMP_POWER_CONTROL, 873
- NG_HCI_LMP_RSSI, 873
- NG_HCI_LMP_SCO_LINK, 874
- NG_HCI_LMP_SLOT_OFFSET, 874
- NG_HCI_LMP_SNIFF_MODE, 874
- NG_HCI_LMP_SWITCH, 874
- NG_HCI_LMP_TIMING_ACCURACY, 874
- NG_HCI_LMP_TRANSPARENT_SCO, 874
- NG_HCI_LMP_ULAW_LOG, 874
- NG_HCI_LOOPBACK_LOCAL, 874
- NG_HCI_LOOPBACK_NONE, 874
- NG_HCI_LOOPBACK_REMOTE, 874
- NG_HCI_MANDATORY_PAGE_SCAN_-
MODE, 874
- NG_HCI_MAX_CON_NUM, 875
- NG_HCI_MAX_NEIGHBOR_NUM, 875
- NG_HCI_MK_CON_HANDLE, 875
- NG_HCI_NO_SCAN_ENABLE, 875
- ng_hci_node_debug_ep, 897
- ng_hci_node_link_policy_mask_ep, 897
- ng_hci_node_packet_mask_ep, 897
- ng_hci_node_role_switch_ep, 897

- ng_hci_node_state_ep, 897
- NG_HCI_NODE_TYPE, 875
- NG_HCI_OCF, 875
- NG_HCI_OCF_ACCEPT_CON, 875
- NG_HCI_OCF_ADD_SCO_CON, 875
- NG_HCI_OCF_AUTH_REQ, 876
- NG_HCI_OCF_CHANGE_CON_LINK_-
KEY, 876
- NG_HCI_OCF_CHANGE_CON_PKT_-
TYPE, 876
- NG_HCI_OCF_CHANGE_LOCAL_NAME,
876
- NG_HCI_OCF_CREATE_CON, 876
- NG_HCI_OCF_CREATE_NEW_UNIT_-
KEY, 876
- NG_HCI_OCF_DELETE_STORED_LINK_-
KEY, 876
- NG_HCI_OCF_DISCON, 876
- NG_HCI_OCF_ENABLE_UNIT_UNDER_-
TEST, 876
- NG_HCI_OCF_EXIT_PARK_MODE, 877
- NG_HCI_OCF_EXIT_PERIODIC_-
INQUIRY, 877
- NG_HCI_OCF_EXIT_SNIFF_MODE, 877
- NG_HCI_OCF_FLUSH, 877
- NG_HCI_OCF_GET_LINK_QUALITY, 877
- NG_HCI_OCF_H2HC_FLOW_CONTROL,
877
- NG_HCI_OCF_HOLD_MODE, 877
- NG_HCI_OCF_HOST_BUFFER_SIZE, 877
- NG_HCI_OCF_HOST_NUM_COMPL_-
PKTS, 878
- NG_HCI_OCF_INQUIRY, 878
- NG_HCI_OCF_INQUIRY_CANCEL, 878
- NG_HCI_OCF_LINK_KEY_NEG_REP, 878
- NG_HCI_OCF_LINK_KEY_REP, 878
- NG_HCI_OCF_MASTER_LINK_KEY, 878
- NG_HCI_OCF_PARK_MODE, 878
- NG_HCI_OCF_PERIODIC_INQUIRY, 878
- NG_HCI_OCF_PIN_CODE_NEG_REP, 878
- NG_HCI_OCF_PIN_CODE_REP, 879
- NG_HCI_OCF_QOS_SETUP, 879
- NG_HCI_OCF_READ_AUTH_ENABLE,
879
- NG_HCI_OCF_READ_AUTO_FLUSH_-
TIMO, 879
- NG_HCI_OCF_READ_BDADDR, 879
- NG_HCI_OCF_READ_BUFFER_SIZE, 879
- NG_HCI_OCF_READ_CLOCK_OFFSET,
879
- NG_HCI_OCF_READ_CON_ACCEPT_-
TIMO, 879
- NG_HCI_OCF_READ_COUNTRY_CODE,
880
- NG_HCI_OCF_READ_ENCRYPTION_-
MODE, 880
- NG_HCI_OCF_READ_FAILED_-
CONTACT_CNTR, 880
- NG_HCI_OCF_READ_HOLD_MODE_-
ACTIVITY, 880
- NG_HCI_OCF_READ_IAC_LAP, 880
- NG_HCI_OCF_READ_INQUIRY_SCAN_-
ACTIVITY, 880
- NG_HCI_OCF_READ_LINK_POLICY_-
SETTINGS, 880
- NG_HCI_OCF_READ_LINK_-
SUPERVISION_TIMO, 880
- NG_HCI_OCF_READ_LOCAL_-
FEATURES, 880
- NG_HCI_OCF_READ_LOCAL_NAME, 881
- NG_HCI_OCF_READ_LOCAL_VER, 881
- NG_HCI_OCF_READ_LOOPBACK_-
MODE, 881
- NG_HCI_OCF_READ_NUM_-
BROADCAST_RETRANS, 881
- NG_HCI_OCF_READ_PAGE_SCAN, 881
- NG_HCI_OCF_READ_PAGE_SCAN_-
ACTIVITY, 881
- NG_HCI_OCF_READ_PAGE_SCAN_-
PERIOD, 881
- NG_HCI_OCF_READ_PAGE_TIMO, 881
- NG_HCI_OCF_READ_PIN_TYPE, 881
- NG_HCI_OCF_READ_REMOTE_-
FEATURES, 882
- NG_HCI_OCF_READ_REMOTE_VER_-
INFO, 882
- NG_HCI_OCF_READ_RSSI, 882
- NG_HCI_OCF_READ_SCAN_ENABLE,
882
- NG_HCI_OCF_READ_SCO_FLOW_-
CONTROL, 882
- NG_HCI_OCF_READ_STORED_LINK_-
KEY, 882
- NG_HCI_OCF_READ_SUPPORTED_IAC_-
NUM, 882
- NG_HCI_OCF_READ_UNIT_CLASS, 882
- NG_HCI_OCF_READ_VOICE_SETTINGS,
883
- NG_HCI_OCF_READ_XMIT_LEVEL, 883
- NG_HCI_OCF_REJECT_CON, 883
- NG_HCI_OCF_REMOTE_NAME_REQ, 883
- NG_HCI_OCF_RESET, 883
- NG_HCI_OCF_RESET_FAILED_-
CONTACT_CNTR, 883
- NG_HCI_OCF_ROLE_DISCOVERY, 883
- NG_HCI_OCF_SET_CON_ENCRYPTION,
883
- NG_HCI_OCF_SET_EVENT_FILTER, 884

- NG_HCI_OCF_SET_EVENT_MASK, 884
- NG_HCI_OCF_SNIFF_MODE, 884
- NG_HCI_OCF_SWITCH_ROLE, 884
- NG_HCI_OCF_WRITE_AUTH_ENABLE, 884
- NG_HCI_OCF_WRITE_AUTO_FLUSH_TIMO, 884
- NG_HCI_OCF_WRITE_CON_ACCEPT_TIMO, 884
- NG_HCI_OCF_WRITE_ENCRYPTION_MODE, 884
- NG_HCI_OCF_WRITE_HOLD_MODE_ACTIVITY, 884
- NG_HCI_OCF_WRITE_IAC_LAP, 885
- NG_HCI_OCF_WRITE_INQUIRY_SCAN_ACTIVITY, 885
- NG_HCI_OCF_WRITE_LINK_POLICY_SETTINGS, 885
- NG_HCI_OCF_WRITE_LINK_SUPERVISION_TIMO, 885
- NG_HCI_OCF_WRITE_LOOPBACK_MODE, 885
- NG_HCI_OCF_WRITE_NUM_BROADCAST_RETRANS, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN_ACTIVITY, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN_PERIOD, 885
- NG_HCI_OCF_WRITE_PAGE_TIMO, 886
- NG_HCI_OCF_WRITE_PIN_TYPE, 886
- NG_HCI_OCF_WRITE_SCAN_ENABLE, 886
- NG_HCI_OCF_WRITE_SCO_FLOW_CONTROL, 886
- NG_HCI_OCF_WRITE_STORED_LINK_KEY, 886
- NG_HCI_OCF_WRITE_UNIT_CLASS, 886
- NG_HCI_OCF_WRITE_VOICE_SETTINGS, 886
- NG_HCI_OGF, 886
- NG_HCI_OGF_BT_LOGO, 887
- NG_HCI_OGF_HC_BASEBAND, 887
- NG_HCI_OGF_INFO, 887
- NG_HCI_OGF_LINK_CONTROL, 887
- NG_HCI_OGF_LINK_POLICY, 887
- NG_HCI_OGF_STATUS, 887
- NG_HCI_OGF_TESTING, 887
- NG_HCI_OGF_VENDOR, 887
- NG_HCI_OPCODE, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE1, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE2, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE3, 888
- NG_HCI_PACKET_FRAGMENT, 888
- NG_HCI_PACKET_START, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE0, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE1, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE2, 888
- NG_HCI_PB_FLAG, 888
- ng_hci_periodic_inquiry_rp, 897
- NG_HCI_PIN_SIZE, 889
- NG_HCI_PIN_TYPE_FIXED, 889
- NG_HCI_PIN_TYPE_VARIABLE, 889
- NG_HCI_PKT_DH1, 889
- NG_HCI_PKT_DH3, 889
- NG_HCI_PKT_DH5, 889
- NG_HCI_PKT_DM1, 889
- NG_HCI_PKT_DM3, 889
- NG_HCI_PKT_DM5, 889
- NG_HCI_PKT_HV1, 890
- NG_HCI_PKT_HV2, 890
- NG_HCI_PKT_HV3, 890
- NG_HCI_POINT2POINT, 890
- ng_hci_reset_rp, 898
- NG_HCI_ROLE_MASTER, 890
- NG_HCI_ROLE_SLAVE, 890
- NG_HCI_SCAN_REP_MODE0, 890
- NG_HCI_SCAN_REP_MODE1, 890
- NG_HCI_SCAN_REP_MODE2, 890
- NG_HCI_SCO_DATA_PKT, 890
- NG_HCI_SCO_PKT_SIZE, 891
- NG_HCI_SERVICE_TYPE_BEST_EFFORT, 891
- NG_HCI_SERVICE_TYPE_GUARANTEED, 891
- NG_HCI_SERVICE_TYPE_NO_TRAFFIC, 891
- ng_hci_set_event_filter_rp, 898
- ng_hci_set_event_mask_rp, 898
- NG_HCI_SPEC_V10, 891
- NG_HCI_SPEC_V11, 891
- NG_HCI_UNIT_COMMAND_PENDING, 891
- NG_HCI_UNIT_CONNECTED, 891
- NG_HCI_UNIT_INITED, 891
- NG_HCI_UNIT_MODE_ACTIVE, 892
- NG_HCI_UNIT_MODE_HOLD, 892
- NG_HCI_UNIT_MODE_PARK, 892
- NG_HCI_UNIT_MODE_SNIFF, 892
- NG_HCI_UNIT_NAME_SIZE, 892
- NG_HCI_UNIT_READY, 892

- NG_HCI_USE_SEMI_PERMANENT_LINK_KEYS, 892
- NG_HCI_USE_TEMPORARY_LINK_KEY, 892
- NG_HCI_WARN_LEVEL, 892
- ng_hci_write_auth_enable_rp, 898
- ng_hci_write_con_accept_timo_rp, 898
- ng_hci_write_encryption_mode_rp, 898
- ng_hci_write_hold_mode_activity_rp, 898
- ng_hci_write_iac_lap_rp, 898
- ng_hci_write_inquiry_scan_activity_rp, 898
- ng_hci_write_loopback_mode_rp, 898
- ng_hci_write_num_broadcast_retrans_rp, 898
- ng_hci_write_page_scan_activity_rp, 899
- ng_hci_write_page_scan_period_rp, 899
- ng_hci_write_page_scan_rp, 899
- ng_hci_write_page_timo_rp, 899
- ng_hci_write_pin_type_rp, 899
- ng_hci_write_scan_enable_rp, 899
- ng_hci_write_sco_flow_control_rp, 899
- ng_hci_write_unit_class_rp, 899
- ng_hci_write_voice_settings_rp, 899
- NG_HCI_XMIT_LEVEL_CURRENT, 892
- NG_HCI_XMIT_LEVEL_MAXIMUM, 893
- NGM_HCI_COOKIE, 893
- NGM_HCI_LP_CON_CFM, 893
- NGM_HCI_LP_CON_IND, 893
- NGM_HCI_LP_CON_REQ, 893
- NGM_HCI_LP_CON_RSP, 893
- NGM_HCI_LP_DISCON_IND, 893
- NGM_HCI_LP_DISCON_REQ, 893
- NGM_HCI_LP_QOS_CFM, 894
- NGM_HCI_LP_QOS_IND, 894
- NGM_HCI_LP_QOS_REQ, 894
- NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE, 894
- NGM_HCI_NODE_GET_BDADDR, 894
- NGM_HCI_NODE_GET_BUFFER, 894
- NGM_HCI_NODE_GET_CON_LIST, 894
- NGM_HCI_NODE_GET_DEBUG, 894
- NGM_HCI_NODE_GET_FEATURES, 894
- NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK, 895
- NGM_HCI_NODE_GET_NEIGHBOR_CACHE, 895
- NGM_HCI_NODE_GET_PACKET_MASK, 895
- NGM_HCI_NODE_GET_ROLE_SWITCH, 895
- NGM_HCI_NODE_GET_STAT, 895
- NGM_HCI_NODE_GET_STATE, 895
- NGM_HCI_NODE_INIT, 895
- NGM_HCI_NODE_LIST_NAMES, 895
- NGM_HCI_NODE_RESET_STAT, 895
- NGM_HCI_NODE_SET_DEBUG, 896
- NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK, 896
- NGM_HCI_NODE_SET_PACKET_MASK, 896
- NGM_HCI_NODE_SET_ROLE_SWITCH, 896
- NGM_HCI_NODE_UP, 896
- NGM_HCI_SYNC_CON_QUEUE, 896
- NG_HCI_ACL_DATA_PKT
 - ng_hci.h, 859
- NG_HCI_ACL_PKT_SIZE
 - ng_hci.h, 859
- ng_hci_acl_rcvdata
 - ng_hci_main.c, 743, 748
- NG_HCI_ALERT
 - ng_hci_var.h, 777
- NG_HCI_ALERT_LEVEL
 - ng_hci.h, 859
- NG_HCI_BC_FLAG
 - ng_hci.h, 859
- NG_HCI_BDADDR_ANY
 - ng_hci.h, 859
- NG_HCI_BDADDR_SIZE
 - ng_hci.h, 859
- ng_hci_bdaddr_type
 - ng_hci_prse.h, 761
- ng_hci_bdaddr_type_info
 - ng_hci_prse.h, 761
- NG_HCI_BROADCAST_ACTIVE
 - ng_hci.h, 859
- NG_HCI_BROADCAST_PICONET
 - ng_hci.h, 859
- NG_HCI_BUFF_ACL_AVAIL
 - ng_hci_var.h, 777
- NG_HCI_BUFF_ACL_FREE
 - ng_hci_var.h, 777
- NG_HCI_BUFF_ACL_SET
 - ng_hci_var.h, 778
- NG_HCI_BUFF_ACL_SIZE
 - ng_hci_var.h, 778
- NG_HCI_BUFF_ACL_TOTAL
 - ng_hci_var.h, 778
- NG_HCI_BUFF_ACL_USE
 - ng_hci_var.h, 778
- NG_HCI_BUFF_CMD_GET
 - ng_hci_var.h, 778
- NG_HCI_BUFF_CMD_SET
 - ng_hci_var.h, 778
- NG_HCI_BUFF_CMD_USE
 - ng_hci_var.h, 778
- NG_HCI_BUFF_SCO_AVAIL
 - ng_hci_var.h, 779
- NG_HCI_BUFF_SCO_FREE

- ng_hci_var.h, 779
- NG_HCI_BUFF_SCO_SET
 - ng_hci_var.h, 779
- NG_HCI_BUFF_SCO_SIZE
 - ng_hci_var.h, 779
- NG_HCI_BUFF_SCO_TOTAL
 - ng_hci_var.h, 779
- NG_HCI_BUFF_SCO_USE
 - ng_hci_var.h, 779
- ng_hci_buffer_type
 - ng_hci_prse.h, 761
- ng_hci_buffer_type_fields
 - ng_hci_prse.h, 762
- ng_hci_change_local_name_rp
 - ng_hci.h, 896
- NG_HCI_CLASS_SIZE
 - ng_hci.h, 859
- NG_HCI_CMD_PKT
 - ng_hci.h, 859
- NG_HCI_CMD_PKT_SIZE
 - ng_hci.h, 860
- NG_HCI_CMD_QUEUE_LEN
 - ng_hci_var.h, 780
- ng_hci_cmdlist
 - ng_hci_prse.h, 762
- ng_hci_cmds.c
 - complete_command, 720
 - min, 720
 - ng_hci_process_command_complete, 720
 - ng_hci_process_command_status, 721
 - ng_hci_process_command_timeout, 721
 - ng_hci_send_command, 722
 - process_hc_baseband_params, 722
 - process_info_params, 723
 - process_link_control_params, 723
 - process_link_control_status, 724
 - process_link_policy_params, 724
 - process_link_policy_status, 724
 - process_status_params, 724
 - process_testing_params, 725
- ng_hci_cmds.h
 - ng_hci_process_command_complete, 726
 - ng_hci_process_command_status, 727
 - ng_hci_process_command_timeout, 727
 - ng_hci_send_command, 728
- ng_hci_command_timeout
 - ng_hci_misc.c, 751
 - ng_hci_misc.h, 756
- ng_hci_command_untimeout
 - ng_hci_misc.c, 751
 - ng_hci_misc.h, 757
- ng_hci_con_by_bdaddr
 - ng_hci_misc.c, 752
 - ng_hci_misc.h, 757
- ng_hci_con_by_handle
 - ng_hci_misc.c, 752
 - ng_hci_misc.h, 757
- NG_HCI_CON_CLOSED
 - ng_hci.h, 860
- NG_HCI_CON_HANDLE
 - ng_hci.h, 860
- NG_HCI_CON_NOTIFY_ACL
 - ng_hci_var.h, 780
- NG_HCI_CON_NOTIFY_SCO
 - ng_hci_var.h, 780
- NG_HCI_CON_OPEN
 - ng_hci.h, 860
- ng_hci_con_timeout
 - ng_hci_misc.c, 752
 - ng_hci_misc.h, 757
- NG_HCI_CON_TIMEOUT_PENDING
 - ng_hci_var.h, 780
- ng_hci_con_untimeout
 - ng_hci_misc.c, 752
 - ng_hci_misc.h, 758
- NG_HCI_CON_W4_CONN_COMPLETE
 - ng_hci.h, 860
- NG_HCI_CON_W4_LP_CON_RSP
 - ng_hci.h, 860
- ng_hci_connect
 - ng_hci_main.c, 743, 748
- ng_hci_constructor
 - ng_hci_main.c, 744, 748
- NG_HCI_COUNTRY_CODE_FRANCE
 - ng_hci.h, 860
- NG_HCI_COUNTRY_CODE_NAM_EUR_JP
 - ng_hci.h, 860
- ng_hci_create_new_unit_key_rp
 - ng_hci.h, 896
- ng_hci_default_rcvmsg
 - ng_hci_main.c, 744, 748
- ng_hci_disconnect
 - ng_hci_main.c, 745, 748
- ng_hci_drv_rcvdata
 - ng_hci_main.c, 745, 748
- ng_hci_enable_unit_under_test_rp
 - ng_hci.h, 897
- NG_HCI_ENCRYPTION_MODE_ALL
 - ng_hci.h, 861
- NG_HCI_ENCRYPTION_MODE_NONE
 - ng_hci.h, 861
- NG_HCI_ENCRYPTION_MODE_P2P
 - ng_hci.h, 861
- NG_HCI_ERR
 - ng_hci_var.h, 780
- NG_HCI_ERR_LEVEL
 - ng_hci.h, 861
- NG_HCI_EVENT_AUTH_COMPL

- ng_hci.h, 861
- NG_HCI_EVENT_BT_LOGO
 - ng_hci.h, 861
- NG_HCI_EVENT_CHANGE_CON_LINK_-
 - KEY_COMPL
 - ng_hci.h, 861
- NG_HCI_EVENT_COMMAND_COMPL
 - ng_hci.h, 861
- NG_HCI_EVENT_COMMAND_STATUS
 - ng_hci.h, 861
- NG_HCI_EVENT_CON_COMPL
 - ng_hci.h, 862
- NG_HCI_EVENT_CON_PKT_TYPE_-
 - CHANGED
 - ng_hci.h, 862
- NG_HCI_EVENT_CON_REQ
 - ng_hci.h, 862
- NG_HCI_EVENT_DATA_BUFFER_OVERFLOW
 - ng_hci.h, 862
- NG_HCI_EVENT_DISCON_COMPL
 - ng_hci.h, 862
- NG_HCI_EVENT_ENCRYPTION_CHANGE
 - ng_hci.h, 862
- NG_HCI_EVENT_FLUSH_OCCUR
 - ng_hci.h, 862
- NG_HCI_EVENT_HARDWARE_ERROR
 - ng_hci.h, 862
- NG_HCI_EVENT_INQUIRY_COMPL
 - ng_hci.h, 862
- NG_HCI_EVENT_INQUIRY_RESULT
 - ng_hci.h, 863
- NG_HCI_EVENT_LINK_KEY_NOTIFICATION
 - ng_hci.h, 863
- NG_HCI_EVENT_LINK_KEY_REQ
 - ng_hci.h, 863
- NG_HCI_EVENT_LOOPBACK_COMMAND
 - ng_hci.h, 863
- NG_HCI_EVENT_MASK_SIZE
 - ng_hci.h, 863
- NG_HCI_EVENT_MASTER_LINK_KEY_-
 - COMPL
 - ng_hci.h, 863
- NG_HCI_EVENT_MAX_SLOT_CHANGE
 - ng_hci.h, 863
- NG_HCI_EVENT_MODE_CHANGE
 - ng_hci.h, 863
- NG_HCI_EVENT_NUM_COMPL_PKTS
 - ng_hci.h, 863
- NG_HCI_EVENT_PAGE_SCAN_MODE_-
 - CHANGE
 - ng_hci.h, 864
- NG_HCI_EVENT_PAGE_SCAN_REP_MODE_-
 - CHANGE
 - ng_hci.h, 864
- NG_HCI_EVENT_PIN_CODE_REQ
 - ng_hci.h, 864
- NG_HCI_EVENT_PKT
 - ng_hci.h, 864
- NG_HCI_EVENT_PKT_SIZE
 - ng_hci.h, 864
- NG_HCI_EVENT_QOS_SETUP_COMPL
 - ng_hci.h, 864
- NG_HCI_EVENT_QOS_VIOLATION
 - ng_hci.h, 864
- NG_HCI_EVENT_READ_CLOCK_OFFSET_-
 - COMPL
 - ng_hci.h, 864
- NG_HCI_EVENT_READ_REMOTE_-
 - FEATURES_COMPL
 - ng_hci.h, 864
- NG_HCI_EVENT_READ_REMOTE_VER_-
 - INFO_COMPL
 - ng_hci.h, 865
- NG_HCI_EVENT_REMOTE_NAME_REQ_-
 - COMPL
 - ng_hci.h, 865
- NG_HCI_EVENT_RETURN_LINK_KEYS
 - ng_hci.h, 865
- NG_HCI_EVENT_ROLE_CHANGE
 - ng_hci.h, 865
- NG_HCI_EVENT_VENDOR
 - ng_hci.h, 865
- NG_HCI_EVMSK_ALL
 - ng_hci.h, 865
- NG_HCI_EVMSK_AUTH_COMPL
 - ng_hci.h, 865
- NG_HCI_EVMSK_CHANGE_CON_LINK_-
 - KEY_COMPL
 - ng_hci.h, 865
- NG_HCI_EVMSK_COMMAND_COMPL
 - ng_hci.h, 865
- NG_HCI_EVMSK_COMMAND_STATUS
 - ng_hci.h, 866
- NG_HCI_EVMSK_CON_COMPL
 - ng_hci.h, 866
- NG_HCI_EVMSK_CON_PKT_TYPE_-
 - CHANGED
 - ng_hci.h, 866
- NG_HCI_EVMSK_CON_REQ
 - ng_hci.h, 866
- NG_HCI_EVMSK_DATA_BUFFER_-
 - OVERFLOW
 - ng_hci.h, 866
- NG_HCI_EVMSK_DISCON_COMPL
 - ng_hci.h, 866
- NG_HCI_EVMSK_ENCRYPTION_CHANGE
 - ng_hci.h, 866
- NG_HCI_EVMSK_FLUSH_OCCUR

- ng_hci.h, 866
- NG_HCI_EVMSK_HARDWARE_ERROR
 - ng_hci.h, 866
- NG_HCI_EVMSK_INQUIRY_COMPL
 - ng_hci.h, 866
- NG_HCI_EVMSK_INQUIRY_RESULT
 - ng_hci.h, 866
- NG_HCI_EVMSK_LINK_KEY_NOTIFICATION
 - ng_hci.h, 867
- NG_HCI_EVMSK_LINK_KEY_REQ
 - ng_hci.h, 867
- NG_HCI_EVMSK_LOOPBACK_COMMAND
 - ng_hci.h, 867
- NG_HCI_EVMSK_MASTER_LINK_KEY_-
COMPL
 - ng_hci.h, 867
- NG_HCI_EVMSK_MAX_SLOT_CHANGE
 - ng_hci.h, 867
- NG_HCI_EVMSK_MODE_CHANGE
 - ng_hci.h, 867
- NG_HCI_EVMSK_NONE
 - ng_hci.h, 867
- NG_HCI_EVMSK_NUM_COMPL_PKTS
 - ng_hci.h, 867
- NG_HCI_EVMSK_PAGE_SCAN_MODE_-
CHANGE
 - ng_hci.h, 867
- NG_HCI_EVMSK_PAGE_SCAN_REP_MODE_-
CHANGE
 - ng_hci.h, 867
- NG_HCI_EVMSK_PIN_CODE_REQ
 - ng_hci.h, 867
- NG_HCI_EVMSK_QOS_SETUP_COMPL
 - ng_hci.h, 868
- NG_HCI_EVMSK_QOS_VIOLATION
 - ng_hci.h, 868
- NG_HCI_EVMSK_READ_CLOCK_OFFSET_-
COMPLETE
 - ng_hci.h, 868
- NG_HCI_EVMSK_READ_REMOTE_-
FEATURES_COMPL
 - ng_hci.h, 868
- NG_HCI_EVMSK_READ_REMOTE_VER_-
INFO_COMPL
 - ng_hci.h, 868
- NG_HCI_EVMSK_REMOTE_NAME_REQ_-
COMPL
 - ng_hci.h, 868
- NG_HCI_EVMSK_RETURN_LINK_KEYS
 - ng_hci.h, 868
- NG_HCI_EVMSK_ROLE_CHANGE
 - ng_hci.h, 868
- ng_hci_evnt.c
 - con_compl, 731
- con_req, 731
- data_buffer_overflow, 732
- discon_compl, 732
- encryption_change, 732
- hardware_error, 733
- inquiry_result, 733
- mode_change, 733
- ng_hci_process_event, 733
- ng_hci_send_data, 735
- num_compl_pkts, 735
- page_scan_mode_change, 735
- page_scan_rep_mode_change, 735
- qos_setup_compl, 736
- qos_violation, 736
- read_clock_offset_compl, 736
- read_remote_features_compl, 737
- role_change, 737
- send_data_packets, 737
- sync_con_queue, 738
- ng_hci_evnt.h
 - ng_hci_process_event, 739
 - ng_hci_send_data, 740
- ng_hci_exit_periodic_inquiry_rp
 - ng_hci.h, 897
- NG_HCI_FEATURES_SIZE
 - ng_hci.h, 868
- ng_hci_features_type
 - ng_hci_prse.h, 762
- ng_hci_features_type_info
 - ng_hci_prse.h, 762
- NG_HCI_FILTER_COND_CON_ANY_UNIT
 - ng_hci.h, 868
- NG_HCI_FILTER_COND_CON_BDADDR
 - ng_hci.h, 868
- NG_HCI_FILTER_COND_CON_UNIT_CLASS
 - ng_hci.h, 869
- NG_HCI_FILTER_COND_INQUIRY_BDADDR
 - ng_hci.h, 869
- NG_HCI_FILTER_COND_INQUIRY_NEW_-
UNIT
 - ng_hci.h, 869
- NG_HCI_FILTER_COND_INQUIRY_UNIT_-
CLASS
 - ng_hci.h, 869
- NG_HCI_FILTER_TYPE_CON_SETUP
 - ng_hci.h, 869
- NG_HCI_FILTER_TYPE_INQUIRY_RESULT
 - ng_hci.h, 869
- NG_HCI_FILTER_TYPE_NONE
 - ng_hci.h, 869
- ng_hci_flush_neighbor_cache
 - ng_hci_misc.c, 753
 - ng_hci_misc.h, 758
- ng_hci_free_con

- ng_hci_misc.c, 753
- ng_hci_misc.h, 758
- ng_hci_free_neighbor
 - ng_hci_misc.c, 753
 - ng_hci_misc.h, 758
- ng_hci_get_neighbor
 - ng_hci_misc.c, 753
 - ng_hci_misc.h, 759
- NG_HCI_H2HC_FLOW_CONTROL_ACL
 - ng_hci.h, 869
- NG_HCI_H2HC_FLOW_CONTROL_BOTH
 - ng_hci.h, 869
- NG_HCI_H2HC_FLOW_CONTROL_NONE
 - ng_hci.h, 869
- ng_hci_h2hc_flow_control_rp
 - ng_hci.h, 897
- NG_HCI_H2HC_FLOW_CONTROL_SCO
 - ng_hci.h, 869
- NG_HCI_HOLD_MODE_NO_CHANGE
 - ng_hci.h, 870
- NG_HCI_HOLD_MODE_SUSPEND_-
INQUIRY_SCAN
 - ng_hci.h, 870
- NG_HCI_HOLD_MODE_SUSPEND_PAGE_-
SCAN
 - ng_hci.h, 870
- NG_HCI_HOLD_MODE_SUSPEND_PERIOD_-
INQUIRY
 - ng_hci.h, 870
- NG_HCI_HOOK_ACL
 - ng_hci.h, 870
- NG_HCI_HOOK_DRV
 - ng_hci.h, 870
- NG_HCI_HOOK_RAW
 - ng_hci.h, 870
- NG_HCI_HOOK_SCO
 - ng_hci.h, 870
- ng_hci_host_buffer_size_rp
 - ng_hci.h, 897
- NG_HCI_INFO
 - ng_hci_var.h, 780
- NG_HCI_INFO_LEVEL
 - ng_hci.h, 870
- ng_hci_inquiry_cancel_rp
 - ng_hci.h, 897
- NG_HCI_INQUIRY_DISABLE_PAGE_ENABLE
 - ng_hci.h, 870
- NG_HCI_INQUIRY_ENABLE_PAGE_DISABLE
 - ng_hci.h, 871
- NG_HCI_INQUIRY_ENABLE_PAGE_ENABLE
 - ng_hci.h, 871
- NG_HCI_KEY_SIZE
 - ng_hci.h, 871
- NG_HCI_LAP_SIZE
 - ng_hci.h, 871
- NG_HCI_LINK_ACL
 - ng_hci.h, 871
- NG_HCI_LINK_KEY_TYPE_COMBINATION_-
KEY
 - ng_hci.h, 871
- NG_HCI_LINK_KEY_TYPE_LOCAL_UNIT_-
KEY
 - ng_hci.h, 871
- NG_HCI_LINK_KEY_TYPE_REMOTE_UNIT_-
KEY
 - ng_hci.h, 871
- NG_HCI_LINK_POLICY_DISABLE_ALL_LM_-
MODES
 - ng_hci.h, 871
- NG_HCI_LINK_POLICY_ENABLE_HOLD_-
MODE
 - ng_hci.h, 871
- NG_HCI_LINK_POLICY_ENABLE_PARK_-
MODE
 - ng_hci.h, 872
- NG_HCI_LINK_POLICY_ENABLE_ROLE_-
SWITCH
 - ng_hci.h, 872
- NG_HCI_LINK_POLICY_ENABLE_SNIFF_-
MODE
 - ng_hci.h, 872
- NG_HCI_LINK_SCO
 - ng_hci.h, 872
- NG_HCI_LMP_3SLOT
 - ng_hci.h, 872
- NG_HCI_LMP_5SLOT
 - ng_hci.h, 872
- NG_HCI_LMP_ALAW_LOG
 - ng_hci.h, 872
- NG_HCI_LMP_CHANNEL_QUALITY
 - ng_hci.h, 872
- NG_HCI_LMP_CVSD
 - ng_hci.h, 872
- NG_HCI_LMP_ENCRYPTION
 - ng_hci.h, 872
- NG_HCI_LMP_FLOW_CONTROL_LAG0
 - ng_hci.h, 873
- NG_HCI_LMP_FLOW_CONTROL_LAG1
 - ng_hci.h, 873
- NG_HCI_LMP_FLOW_CONTROL_LAG2
 - ng_hci.h, 873
- NG_HCI_LMP_HOLD_MODE
 - ng_hci.h, 873
- NG_HCI_LMP_HV2_PKT
 - ng_hci.h, 873
- NG_HCI_LMP_HV3_PKT
 - ng_hci.h, 873
- NG_HCI_LMP_PAGING_SCHEME
 - ng_hci.h, 873

- ng_hci.h, 873
- NG_HCI_LMP_PARK_MODE
 - ng_hci.h, 873
- NG_HCI_LMP_POWER_CONTROL
 - ng_hci.h, 873
- NG_HCI_LMP_RSSI
 - ng_hci.h, 873
- NG_HCI_LMP_SCO_LINK
 - ng_hci.h, 874
- NG_HCI_LMP_SLOT_OFFSET
 - ng_hci.h, 874
- NG_HCI_LMP_SNIFF_MODE
 - ng_hci.h, 874
- NG_HCI_LMP_SWITCH
 - ng_hci.h, 874
- NG_HCI_LMP_TIMING_ACCURACY
 - ng_hci.h, 874
- NG_HCI_LMP_TRANSPARENT_SCO
 - ng_hci.h, 874
- NG_HCI_LMP_ULAW_LOG
 - ng_hci.h, 874
- NG_HCI_LOOPBACK_LOCAL
 - ng_hci.h, 874
- NG_HCI_LOOPBACK_NONE
 - ng_hci.h, 874
- NG_HCI_LOOPBACK_REMOTE
 - ng_hci.h, 874
- ng_hci_lp_acl_con_req
 - ng_hci_ulpi.c, 765
- ng_hci_lp_con_cfm
 - ng_hci_ulpi.c, 766
 - ng_hci_ulpi.h, 771
- ng_hci_lp_con_cfm_ep, 212
 - bdaddr, 212
 - con_handle, 212
 - link_type, 212
 - status, 212
- ng_hci_lp_con_ind
 - ng_hci_ulpi.c, 766
 - ng_hci_ulpi.h, 771
- ng_hci_lp_con_ind_ep, 213
 - bdaddr, 213
 - link_type, 213
 - uclass, 213
- ng_hci_lp_con_req
 - ng_hci_ulpi.c, 767
 - ng_hci_ulpi.h, 771
- ng_hci_lp_con_req_ep, 214
 - bdaddr, 214
 - link_type, 214
- ng_hci_lp_con_rsp
 - ng_hci_ulpi.c, 767
 - ng_hci_ulpi.h, 772
- ng_hci_lp_con_rsp_ep, 215
 - bdaddr, 215
 - link_type, 215
 - status, 215
- ng_hci_lp_discon_ind
 - ng_hci_ulpi.c, 768
 - ng_hci_ulpi.h, 773
- ng_hci_lp_discon_ind_ep, 216
 - con_handle, 216
 - link_type, 216
 - reason, 216
- ng_hci_lp_discon_req
 - ng_hci_ulpi.c, 768
 - ng_hci_ulpi.h, 773
- ng_hci_lp_discon_req_ep, 217
 - con_handle, 217
 - reason, 217
- ng_hci_lp_qos_cfm
 - ng_hci_ulpi.c, 768
 - ng_hci_ulpi.h, 773
- ng_hci_lp_qos_cfm_ep, 218
 - con_handle, 218
 - status, 218
- ng_hci_lp_qos_ind
 - ng_hci_ulpi.c, 769
 - ng_hci_ulpi.h, 774
- ng_hci_lp_qos_ind_ep, 219
 - con_handle, 219
- ng_hci_lp_qos_req
 - ng_hci_ulpi.c, 769
 - ng_hci_ulpi.h, 774
- ng_hci_lp_qos_req_ep, 220
 - con_handle, 220
 - delay_variation, 220
 - flags, 220
 - latency, 220
 - peak_bandwidth, 220
 - service_type, 220
 - token_rate, 220
- ng_hci_lp_sco_con_req
 - ng_hci_ulpi.c, 769
- NG_HCI_M_PULLUP
 - ng_hci_var.h, 780
- ng_hci_main.c
 - M_NETGRAPH_HCI, 743
 - MODULE_DEPEND, 743
 - MODULE_VERSION, 743
 - NETGRAPH_INIT, 743
 - ng_hci_acl_rcvdata, 743, 748
 - ng_hci_connect, 743, 748
 - ng_hci_constructor, 744, 748
 - ng_hci_default_rcvmsg, 744, 748
 - ng_hci_disconnect, 745, 748
 - ng_hci_drv_rcvdata, 745, 748
 - ng_hci_newhook, 746, 749

- ng_hci_raw_rcvdata, 746, 749
- ng_hci_sco_rcvdata, 747, 749
- ng_hci_shutdown, 747, 749
- ng_hci_upper_rcvmsg, 747, 749
- typestruct, 749
- NG_HCI_MANDATORY_PAGE_SCAN_MODE
 - ng_hci.h, 874
- NG_HCI_MAX_CON_NUM
 - ng_hci.h, 875
- NG_HCI_MAX_NEIGHBOR_NUM
 - ng_hci.h, 875
- ng_hci_misc.c
 - ng_hci_command_timeout, 751
 - ng_hci_command_untimeout, 751
 - ng_hci_con_by_bdaddr, 752
 - ng_hci_con_by_handle, 752
 - ng_hci_con_timeout, 752
 - ng_hci_con_untimeout, 752
 - ng_hci_flush_neighbor_cache, 753
 - ng_hci_free_con, 753
 - ng_hci_free_neighbor, 753
 - ng_hci_get_neighbor, 753
 - ng_hci_mtap, 753
 - ng_hci_neighbor_stale, 754
 - ng_hci_new_con, 754
 - ng_hci_new_neighbor, 754
 - ng_hci_node_is_up, 754
 - ng_hci_unit_clean, 754
- ng_hci_misc.h
 - ng_hci_command_timeout, 756
 - ng_hci_command_untimeout, 757
 - ng_hci_con_by_bdaddr, 757
 - ng_hci_con_by_handle, 757
 - ng_hci_con_timeout, 757
 - ng_hci_con_untimeout, 758
 - ng_hci_flush_neighbor_cache, 758
 - ng_hci_free_con, 758
 - ng_hci_free_neighbor, 758
 - ng_hci_get_neighbor, 759
 - ng_hci_mtap, 759
 - ng_hci_neighbor_stale, 759
 - ng_hci_new_con, 759
 - ng_hci_new_neighbor, 759
 - ng_hci_node_is_up, 760
 - ng_hci_unit_clean, 760
- NG_HCI_MK_CON_HANDLE
 - ng_hci.h, 875
- ng_hci_mtap
 - ng_hci_misc.c, 753
 - ng_hci_misc.h, 759
- ng_hci_neighbor, 222
 - bdaddr, 222
 - clock_offset, 222
 - features, 222
 - LIST_ENTRY, 222
 - page_scan_mode, 222
 - page_scan_rep_mode, 223
 - updated, 223
- ng_hci_neighbor_p
 - ng_hci_var.h, 782
- ng_hci_neighbor_stale
 - ng_hci_misc.c, 754
 - ng_hci_misc.h, 759
- ng_hci_neighbor_t
 - ng_hci_var.h, 782
- ng_hci_new_con
 - ng_hci_misc.c, 754
 - ng_hci_misc.h, 759
- ng_hci_new_neighbor
 - ng_hci_misc.c, 754
 - ng_hci_misc.h, 759
- ng_hci_newhook
 - ng_hci_main.c, 746, 749
- NG_HCI_NO_SCAN_ENABLE
 - ng_hci.h, 875
- ng_hci_node_buffer_ep, 224
 - acl_free, 224
 - acl_pkts, 224
 - acl_size, 224
 - cmd_free, 224
 - sco_free, 224
 - sco_pkts, 224
 - sco_size, 225
- ng_hci_node_con_ep, 226
 - bdaddr, 226
 - con_handle, 226
 - encryption_mode, 226
 - link_type, 226
 - mode, 226
 - pending, 226
 - queue_len, 226
 - reserved, 227
 - role, 227
 - state, 227
- ng_hci_node_con_list_ep, 228
 - num_connections, 228
- ng_hci_node_debug_ep
 - ng_hci.h, 897
- ng_hci_node_get_neighbor_cache_ep, 229
 - num_entries, 229
- ng_hci_node_is_up
 - ng_hci_misc.c, 754
 - ng_hci_misc.h, 760
- ng_hci_node_link_policy_mask_ep
 - ng_hci.h, 897
- ng_hci_node_neighbor_cache_entry_ep, 230
 - bdaddr, 230
 - clock_offset, 230

- features, 230
- page_scan_mode, 230
- page_scan_rep_mode, 230
- ng_hci_node_packet_mask_ep
 - ng_hci.h, 897
- ng_hci_node_role_switch_ep
 - ng_hci.h, 897
- ng_hci_node_stat_ep, 231
 - acl_recv, 231
 - acl_sent, 231
 - bytes_recv, 231
 - bytes_sent, 231
 - cmd_sent, 231
 - evnt_recv, 231
 - sco_recv, 231
 - sco_sent, 232
- ng_hci_node_state_ep
 - ng_hci.h, 897
- NG_HCI_NODE_TYPE
 - ng_hci.h, 875
- ng_hci_node_up_ep, 233
 - bdaddr, 233
 - num_pkts, 233
 - pkt_size, 233
 - reserved, 233
- NG_HCI_OCF
 - ng_hci.h, 875
- NG_HCI_OCF_ACCEPT_CON
 - ng_hci.h, 875
- NG_HCI_OCF_ADD_SCO_CON
 - ng_hci.h, 875
- NG_HCI_OCF_AUTH_REQ
 - ng_hci.h, 876
- NG_HCI_OCF_CHANGE_CON_LINK_KEY
 - ng_hci.h, 876
- NG_HCI_OCF_CHANGE_CON_PKT_TYPE
 - ng_hci.h, 876
- NG_HCI_OCF_CHANGE_LOCAL_NAME
 - ng_hci.h, 876
- NG_HCI_OCF_CREATE_CON
 - ng_hci.h, 876
- NG_HCI_OCF_CREATE_NEW_UNIT_KEY
 - ng_hci.h, 876
- NG_HCI_OCF_DELETE_STORED_LINK_KEY
 - ng_hci.h, 876
- NG_HCI_OCF_DISCON
 - ng_hci.h, 876
- NG_HCI_OCF_ENABLE_UNIT_UNDER_TEST
 - ng_hci.h, 876
- NG_HCI_OCF_EXIT_PARK_MODE
 - ng_hci.h, 877
- NG_HCI_OCF_EXIT_PERIODIC_INQUIRY
 - ng_hci.h, 877
- NG_HCI_OCF_EXIT_SNIFF_MODE
 - ng_hci.h, 877
- NG_HCI_OCF_FLUSH
 - ng_hci.h, 877
- NG_HCI_OCF_GET_LINK_QUALITY
 - ng_hci.h, 877
- NG_HCI_OCF_H2HC_FLOW_CONTROL
 - ng_hci.h, 877
- NG_HCI_OCF_HOLD_MODE
 - ng_hci.h, 877
- NG_HCI_OCF_HOST_BUFFER_SIZE
 - ng_hci.h, 877
- NG_HCI_OCF_HOST_NUM_COMPL_PKTS
 - ng_hci.h, 878
- NG_HCI_OCF_INQUIRY
 - ng_hci.h, 878
- NG_HCI_OCF_INQUIRY_CANCEL
 - ng_hci.h, 878
- NG_HCI_OCF_LINK_KEY_NEG_REP
 - ng_hci.h, 878
- NG_HCI_OCF_LINK_KEY_REP
 - ng_hci.h, 878
- NG_HCI_OCF_MASTER_LINK_KEY
 - ng_hci.h, 878
- NG_HCI_OCF_PARK_MODE
 - ng_hci.h, 878
- NG_HCI_OCF_PERIODIC_INQUIRY
 - ng_hci.h, 878
- NG_HCI_OCF_PIN_CODE_NEG_REP
 - ng_hci.h, 878
- NG_HCI_OCF_PIN_CODE_REP
 - ng_hci.h, 879
- NG_HCI_OCF_QOS_SETUP
 - ng_hci.h, 879
- NG_HCI_OCF_READ_AUTH_ENABLE
 - ng_hci.h, 879
- NG_HCI_OCF_READ_AUTO_FLUSH_TIMO
 - ng_hci.h, 879
- NG_HCI_OCF_READ_BDADDR
 - ng_hci.h, 879
- NG_HCI_OCF_READ_BUFFER_SIZE
 - ng_hci.h, 879
- NG_HCI_OCF_READ_CLOCK_OFFSET
 - ng_hci.h, 879
- NG_HCI_OCF_READ_CON_ACCEPT_TIMO
 - ng_hci.h, 879
- NG_HCI_OCF_READ_COUNTRY_CODE
 - ng_hci.h, 880
- NG_HCI_OCF_READ_ENCRYPTION_MODE
 - ng_hci.h, 880
- NG_HCI_OCF_READ_FAILED_CONTACT_-
CNTR
 - ng_hci.h, 880
- NG_HCI_OCF_READ_HOLD_MODE_-
ACTIVITY
 - ng_hci.h, 880

- ng_hci.h, 880
- NG_HCI_OCF_READ_IAC_LAP
 - ng_hci.h, 880
- NG_HCI_OCF_READ_INQUIRY_SCAN_-
 - ACTIVITY
 - ng_hci.h, 880
- NG_HCI_OCF_READ_LINK_POLICY_-
 - SETTINGS
 - ng_hci.h, 880
- NG_HCI_OCF_READ_LINK_SUPERVISION_-
 - TIMO
 - ng_hci.h, 880
- NG_HCI_OCF_READ_LOCAL_FEATURES
 - ng_hci.h, 880
- NG_HCI_OCF_READ_LOCAL_NAME
 - ng_hci.h, 881
- NG_HCI_OCF_READ_LOCAL_VER
 - ng_hci.h, 881
- NG_HCI_OCF_READ_LOOPBACK_MODE
 - ng_hci.h, 881
- NG_HCI_OCF_READ_NUM_BROADCAST_-
 - RETRANS
 - ng_hci.h, 881
- NG_HCI_OCF_READ_PAGE_SCAN
 - ng_hci.h, 881
- NG_HCI_OCF_READ_PAGE_SCAN_ACTIVITY
 - ng_hci.h, 881
- NG_HCI_OCF_READ_PAGE_SCAN_PERIOD
 - ng_hci.h, 881
- NG_HCI_OCF_READ_PAGE_TIMO
 - ng_hci.h, 881
- NG_HCI_OCF_READ_PIN_TYPE
 - ng_hci.h, 881
- NG_HCI_OCF_READ_REMOTE_FEATURES
 - ng_hci.h, 882
- NG_HCI_OCF_READ_REMOTE_VER_INFO
 - ng_hci.h, 882
- NG_HCI_OCF_READ_RSSI
 - ng_hci.h, 882
- NG_HCI_OCF_READ_SCAN_ENABLE
 - ng_hci.h, 882
- NG_HCI_OCF_READ_SCO_FLOW_CONTROL
 - ng_hci.h, 882
- NG_HCI_OCF_READ_STORED_LINK_KEY
 - ng_hci.h, 882
- NG_HCI_OCF_READ_SUPPORTED_IAC_NUM
 - ng_hci.h, 882
- NG_HCI_OCF_READ_UNIT_CLASS
 - ng_hci.h, 882
- NG_HCI_OCF_READ_VOICE_SETTINGS
 - ng_hci.h, 883
- NG_HCI_OCF_READ_XMIT_LEVEL
 - ng_hci.h, 883
- NG_HCI_OCF_REJECT_CON
 - ng_hci.h, 883
- NG_HCI_OCF_REMOTE_NAME_REQ
 - ng_hci.h, 883
- NG_HCI_OCF_RESET
 - ng_hci.h, 883
- NG_HCI_OCF_RESET_FAILED_CONTACT_-
 - CNTR
 - ng_hci.h, 883
- NG_HCI_OCF_ROLE_DISCOVERY
 - ng_hci.h, 883
- NG_HCI_OCF_SET_CON_ENCRYPTION
 - ng_hci.h, 883
- NG_HCI_OCF_SET_EVENT_FILTER
 - ng_hci.h, 884
- NG_HCI_OCF_SET_EVENT_MASK
 - ng_hci.h, 884
- NG_HCI_OCF_SNIFF_MODE
 - ng_hci.h, 884
- NG_HCI_OCF_SWITCH_ROLE
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_AUTH_ENABLE
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_AUTO_FLUSH_TIMO
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_CON_ACCEPT_TIMO
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_ENCRYPTION_MODE
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_HOLD_MODE_-
 - ACTIVITY
 - ng_hci.h, 884
- NG_HCI_OCF_WRITE_IAC_LAP
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_INQUIRY_SCAN_-
 - ACTIVITY
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_LINK_POLICY_-
 - SETTINGS
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_LINK_SUPERVISION_-
 - TIMO
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_LOOPBACK_MODE
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_NUM_BROADCAST_-
 - RETRANS
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN_-
 - ACTIVITY
 - ng_hci.h, 885
- NG_HCI_OCF_WRITE_PAGE_SCAN_PERIOD
 - ng_hci.h, 885

- NG_HCI_OCF_WRITE_PAGE_TIMO
ng_hci.h, 886
- NG_HCI_OCF_WRITE_PIN_TYPE
ng_hci.h, 886
- NG_HCI_OCF_WRITE_SCAN_ENABLE
ng_hci.h, 886
- NG_HCI_OCF_WRITE_SCO_FLOW_CONTROL
ng_hci.h, 886
- NG_HCI_OCF_WRITE_STORED_LINK_KEY
ng_hci.h, 886
- NG_HCI_OCF_WRITE_UNIT_CLASS
ng_hci.h, 886
- NG_HCI_OCF_WRITE_VOICE_SETTINGS
ng_hci.h, 886
- NG_HCI_OGF
ng_hci.h, 886
- NG_HCI_OGF_BT_LOGO
ng_hci.h, 887
- NG_HCI_OGF_HC_BASEBAND
ng_hci.h, 887
- NG_HCI_OGF_INFO
ng_hci.h, 887
- NG_HCI_OGF_LINK_CONTROL
ng_hci.h, 887
- NG_HCI_OGF_LINK_POLICY
ng_hci.h, 887
- NG_HCI_OGF_STATUS
ng_hci.h, 887
- NG_HCI_OGF_TESTING
ng_hci.h, 887
- NG_HCI_OGF_VENDOR
ng_hci.h, 887
- NG_HCI_OPCODE
ng_hci.h, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE1
ng_hci.h, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE2
ng_hci.h, 888
- NG_HCI_OPTIONAL_PAGE_SCAN_MODE3
ng_hci.h, 888
- NG_HCI_PACKET_FRAGMENT
ng_hci.h, 888
- NG_HCI_PACKET_START
ng_hci.h, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE0
ng_hci.h, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE1
ng_hci.h, 888
- NG_HCI_PAGE_SCAN_PERIOD_MODE2
ng_hci.h, 888
- NG_HCI_PB_FLAG
ng_hci.h, 888
- ng_hci_periodic_inquiry_rp
ng_hci.h, 897
- NG_HCI_PIN_SIZE
ng_hci.h, 889
- NG_HCI_PIN_TYPE_FIXED
ng_hci.h, 889
- NG_HCI_PIN_TYPE_VARIABLE
ng_hci.h, 889
- NG_HCI_PKT_DH1
ng_hci.h, 889
- NG_HCI_PKT_DH3
ng_hci.h, 889
- NG_HCI_PKT_DH5
ng_hci.h, 889
- NG_HCI_PKT_DM1
ng_hci.h, 889
- NG_HCI_PKT_DM3
ng_hci.h, 889
- NG_HCI_PKT_DM5
ng_hci.h, 889
- NG_HCI_PKT_HV1
ng_hci.h, 890
- NG_HCI_PKT_HV2
ng_hci.h, 890
- NG_HCI_PKT_HV3
ng_hci.h, 890
- NG_HCI_POINT2POINT
ng_hci.h, 890
- ng_hci_process_command_complete
ng_hci_cmds.c, 720
ng_hci_cmds.h, 726
- ng_hci_process_command_status
ng_hci_cmds.c, 721
ng_hci_cmds.h, 727
- ng_hci_process_command_timeout
ng_hci_cmds.c, 721
ng_hci_cmds.h, 727
- ng_hci_process_con_timeout
ng_hci_ulpi.c, 770
ng_hci_ulpi.h, 774
- ng_hci_process_event
ng_hci_evnt.c, 733
ng_hci_evnt.h, 739
- ng_hci_prse.h
ng_hci_bdaddr_type, 761
ng_hci_bdaddr_type_info, 761
ng_hci_buffer_type, 761
ng_hci_buffer_type_fields, 762
ng_hci_cmdlist, 762
ng_hci_features_type, 762
ng_hci_features_type_info, 762
ng_hci_stat_type, 762
ng_hci_stat_type_fields, 763
- ng_hci_raw_rcvdata
ng_hci_main.c, 746, 749
- ng_hci_reset_rp

- ng_hci.h, 898
- NG_HCI_ROLE_MASTER
 - ng_hci.h, 890
- NG_HCI_ROLE_SLAVE
 - ng_hci.h, 890
- NG_HCI_SCAN_REP_MODE0
 - ng_hci.h, 890
- NG_HCI_SCAN_REP_MODE1
 - ng_hci.h, 890
- NG_HCI_SCAN_REP_MODE2
 - ng_hci.h, 890
- NG_HCI_SCO_DATA_PKT
 - ng_hci.h, 890
- NG_HCI_SCO_PKT_SIZE
 - ng_hci.h, 891
- ng_hci_sco_rcvdata
 - ng_hci_main.c, 747, 749
- ng_hci_send_command
 - ng_hci_cmds.c, 722
 - ng_hci_cmds.h, 728
- ng_hci_send_data
 - ng_hci_evnt.c, 735
 - ng_hci_evnt.h, 740
- NG_HCI_SERVICE_TYPE_BEST_EFFORT
 - ng_hci.h, 891
- NG_HCI_SERVICE_TYPE_GUARANTEED
 - ng_hci.h, 891
- NG_HCI_SERVICE_TYPE_NO_TRAFFIC
 - ng_hci.h, 891
- ng_hci_set_event_filter_rp
 - ng_hci.h, 898
- ng_hci_set_event_mask_rp
 - ng_hci.h, 898
- ng_hci_shutdown
 - ng_hci_main.c, 747, 749
- NG_HCI_SPEC_V10
 - ng_hci.h, 891
- NG_HCI_SPEC_V11
 - ng_hci.h, 891
- NG_HCI_STAT_ACL_RECV
 - ng_hci_var.h, 781
- NG_HCI_STAT_ACL_SENT
 - ng_hci_var.h, 781
- NG_HCI_STAT_BYTES_RECV
 - ng_hci_var.h, 781
- NG_HCI_STAT_BYTES_SENT
 - ng_hci_var.h, 781
- NG_HCI_STAT_CMD_SENT
 - ng_hci_var.h, 781
- NG_HCI_STAT_EVTN_RECV
 - ng_hci_var.h, 781
- NG_HCI_STAT_RESET
 - ng_hci_var.h, 781
- NG_HCI_STAT_SCO_RECV
 - ng_hci_var.h, 782
- NG_HCI_STAT_SCO_SENT
 - ng_hci_var.h, 782
- ng_hci_stat_type
 - ng_hci_prse.h, 762
- ng_hci_stat_type_fields
 - ng_hci_prse.h, 763
- ng_hci_sync_con_queue_ep, 234
 - completed, 234
 - con_handle, 234
- ng_hci_ulpi.c
 - ng_hci_lp_acl_con_req, 765
 - ng_hci_lp_con_cfm, 766
 - ng_hci_lp_con_ind, 766
 - ng_hci_lp_con_req, 767
 - ng_hci_lp_con_rsp, 767
 - ng_hci_lp_discon_ind, 768
 - ng_hci_lp_discon_req, 768
 - ng_hci_lp_qos_cfm, 768
 - ng_hci_lp_qos_ind, 769
 - ng_hci_lp_qos_req, 769
 - ng_hci_lp_sco_con_req, 769
 - ng_hci_process_con_timeout, 770
- ng_hci_ulpi.h
 - ng_hci_lp_con_cfm, 771
 - ng_hci_lp_con_ind, 771
 - ng_hci_lp_con_req, 771
 - ng_hci_lp_con_rsp, 772
 - ng_hci_lp_discon_ind, 773
 - ng_hci_lp_discon_req, 773
 - ng_hci_lp_qos_cfm, 773
 - ng_hci_lp_qos_ind, 774
 - ng_hci_lp_qos_req, 774
 - ng_hci_process_con_timeout, 774
- ng_hci_unit, 235
 - acl, 236
 - bdaddr, 236
 - buffer, 236
 - cmd_timo, 236
 - cmdq, 236
 - debug, 236
 - drv, 236
 - features, 237
 - link_policy_mask, 237
 - LIST_HEAD, 236
 - node, 237
 - packet_mask, 237
 - raw, 237
 - role_switch, 237
 - sco, 238
 - stat, 238
 - state, 238
- ng_hci_unit_buff, 239
 - acl_free, 239

- acl_pkts, [239](#)
- acl_size, [239](#)
- cmd_free, [239](#)
- sco_free, [239](#)
- sco_pkts, [239](#)
- sco_size, [239](#)
- ng_hci_unit_buff_t
 - ng_hci_var.h, [782](#)
- ng_hci_unit_clean
 - ng_hci_misc.c, [754](#)
 - ng_hci_misc.h, [760](#)
- NG_HCI_UNIT_COMMAND_PENDING
 - ng_hci.h, [891](#)
- ng_hci_unit_con, [241](#)
 - bdaddr, [242](#)
 - con_handle, [242](#)
 - con_timo, [242](#)
 - conq, [242](#)
 - encryption_mode, [242](#)
 - flags, [242](#)
 - link_type, [242](#)
 - LIST_ENTRY, [242](#)
 - mode, [243](#)
 - pending, [243](#)
 - role, [243](#)
 - state, [243](#)
 - unit, [243](#)
- ng_hci_unit_con_p
 - ng_hci_var.h, [782](#)
- ng_hci_unit_con_t
 - ng_hci_var.h, [782](#)
- NG_HCI_UNIT_CONNECTED
 - ng_hci.h, [891](#)
- NG_HCI_UNIT_INITED
 - ng_hci.h, [891](#)
- NG_HCI_UNIT_MODE_ACTIVE
 - ng_hci.h, [892](#)
- NG_HCI_UNIT_MODE_HOLD
 - ng_hci.h, [892](#)
- NG_HCI_UNIT_MODE_PARK
 - ng_hci.h, [892](#)
- NG_HCI_UNIT_MODE_SNIFF
 - ng_hci.h, [892](#)
- NG_HCI_UNIT_NAME_SIZE
 - ng_hci.h, [892](#)
- ng_hci_unit_p
 - ng_hci_var.h, [782](#)
- NG_HCI_UNIT_READY
 - ng_hci.h, [892](#)
- ng_hci_unit_t
 - ng_hci_var.h, [782](#)
- ng_hci_upper_rcvmsg
 - ng_hci_main.c, [747](#), [749](#)
- NG_HCI_USE_SEMI_PERMANENT_LINK_KEYS
 - ng_hci.h, [892](#)
- NG_HCI_USE_TEMPORARY_LINK_KEY
 - ng_hci.h, [892](#)
- ng_hci_var.h
 - M_NETGRAPH_HCI, [777](#)
 - NG_HCI_ALERT, [777](#)
 - NG_HCI_BUFF_ACL_AVAIL, [777](#)
 - NG_HCI_BUFF_ACL_FREE, [777](#)
 - NG_HCI_BUFF_ACL_SET, [778](#)
 - NG_HCI_BUFF_ACL_SIZE, [778](#)
 - NG_HCI_BUFF_ACL_TOTAL, [778](#)
 - NG_HCI_BUFF_ACL_USE, [778](#)
 - NG_HCI_BUFF_CMD_GET, [778](#)
 - NG_HCI_BUFF_CMD_SET, [778](#)
 - NG_HCI_BUFF_CMD_USE, [778](#)
 - NG_HCI_BUFF_SCO_AVAIL, [779](#)
 - NG_HCI_BUFF_SCO_FREE, [779](#)
 - NG_HCI_BUFF_SCO_SET, [779](#)
 - NG_HCI_BUFF_SCO_SIZE, [779](#)
 - NG_HCI_BUFF_SCO_TOTAL, [779](#)
 - NG_HCI_BUFF_SCO_USE, [779](#)
 - NG_HCI_CMD_QUEUE_LEN, [780](#)
 - NG_HCI_CON_NOTIFY_ACL, [780](#)
 - NG_HCI_CON_NOTIFY_SCO, [780](#)
 - NG_HCI_CON_TIMEOUT_PENDING, [780](#)
 - NG_HCI_ERR, [780](#)
 - NG_HCI_INFO, [780](#)
 - NG_HCI_M_PULLUP, [780](#)
 - ng_hci_neighbor_p, [782](#)
 - ng_hci_neighbor_t, [782](#)
 - NG_HCI_STAT_ACL_RECV, [781](#)
 - NG_HCI_STAT_ACL_SENT, [781](#)
 - NG_HCI_STAT_BYTES_RECV, [781](#)
 - NG_HCI_STAT_BYTES_SENT, [781](#)
 - NG_HCI_STAT_CMD_SENT, [781](#)
 - NG_HCI_STAT_EVNT_RECV, [781](#)
 - NG_HCI_STAT_RESET, [781](#)
 - NG_HCI_STAT_SCO_RECV, [782](#)
 - NG_HCI_STAT_SCO_SENT, [782](#)
 - ng_hci_unit_buff_t, [782](#)
 - ng_hci_unit_con_p, [782](#)
 - ng_hci_unit_con_t, [782](#)
 - ng_hci_unit_p, [782](#)
 - ng_hci_unit_t, [782](#)
 - NG_HCI_WARN, [782](#)
- NG_HCI_WARN
 - ng_hci_var.h, [782](#)
- NG_HCI_WARN_LEVEL
 - ng_hci.h, [892](#)
- ng_hci_write_auth_enable_rp
 - ng_hci.h, [898](#)
- ng_hci_write_con_accept_timo_rp

- ng_hci.h, 898
- ng_hci_write_encryption_mode_rp
 - ng_hci.h, 898
- ng_hci_write_hold_mode_activity_rp
 - ng_hci.h, 898
- ng_hci_write_iac_lap_rp
 - ng_hci.h, 898
- ng_hci_write_inquiry_scan_activity_rp
 - ng_hci.h, 898
- ng_hci_write_loopback_mode_rp
 - ng_hci.h, 898
- ng_hci_write_num_broadcast_retrans_rp
 - ng_hci.h, 898
- ng_hci_write_page_scan_activity_rp
 - ng_hci.h, 899
- ng_hci_write_page_scan_period_rp
 - ng_hci.h, 899
- ng_hci_write_page_scan_rp
 - ng_hci.h, 899
- ng_hci_write_page_timo_rp
 - ng_hci.h, 899
- ng_hci_write_pin_type_rp
 - ng_hci.h, 899
- ng_hci_write_scan_enable_rp
 - ng_hci.h, 899
- ng_hci_write_sco_flow_control_rp
 - ng_hci.h, 899
- ng_hci_write_unit_class_rp
 - ng_hci.h, 899
- ng_hci_write_voice_settings_rp
 - ng_hci.h, 899
- NG_HCI_XMIT_LEVEL_CURRENT
 - ng_hci.h, 892
- NG_HCI_XMIT_LEVEL_MAXIMUM
 - ng_hci.h, 893
- ng_hole.c
 - hinfo_p, 1327
 - NETGRAPH_INIT, 1327
 - ng_hole_cmdlist, 1328
 - ng_hole_hookstat_type, 1328
 - ng_hole_hookstat_type_fields, 1329
 - ngh_cons, 1327, 1329
 - ngh_disconnect, 1327, 1329
 - ngh_newhook, 1327, 1329
 - ngh_rcvdata, 1327, 1329
 - ngh_rcvmsg, 1328, 1329
 - typestruct, 1329
- ng_hole.h
 - NGM_HOLE_CLR_STATS, 1331
 - NGM_HOLE_GET_STATS, 1331
 - NGM_HOLE_GETCLR_STATS, 1331
- ng_hole.h
 - NG_HOLE_HOOKSTAT_TYPE_INFO, 1330
 - NG_HOLE_NODE_TYPE, 1330
 - NGM_HOLE_COOKIE, 1330
- ng_hole_cmdlist
 - ng_hole.c, 1328
- ng_hole_hookinfo, 244
 - stats, 244
- ng_hole_hookstat, 245
 - frames, 245
 - octets, 245
- ng_hole_hookstat_type
 - ng_hole.c, 1328
- ng_hole_hookstat_type_fields
 - ng_hole.c, 1329
- NG_HOLE_HOOKSTAT_TYPE_INFO
 - ng_hole.h, 1330
- NG_HOLE_NODE_TYPE
 - ng_hole.h, 1330
- ng_hook, 246
 - hk_flags, 247
 - hk_name, 247
 - hk_node, 247
 - hk_peer, 247
 - hk_private, 247
 - hk_rcvdata, 247
 - hk_rcvmsg, 247
 - hk_refs, 247
 - hk_type, 247
 - LIST_ENTRY, 247
- NG_HOOK_FORCE_QUEUE
 - netgraph.h, 1131
- NG_HOOK_FORCE_WRITER
 - netgraph.h, 1131
- NG_HOOK_IS_VALID
 - netgraph.h, 1132
- NG_HOOK_NAME
 - netgraph.h, 1132
- NG_HOOK_NODE
 - netgraph.h, 1132
- NG_HOOK_NOT_VALID
 - netgraph.h, 1133
- NG_HOOK_PEER
 - netgraph.h, 1133
- NG_HOOK_PRIVATE
 - netgraph.h, 1133
- NG_HOOK_REF
 - netgraph.h, 1133
- NG_HOOK_SET_PRIVATE
 - netgraph.h, 1134
- NG_HOOK_SET_RCVDATA
 - netgraph.h, 1134
- NG_HOOK_SET_RCVMSG
 - netgraph.h, 1134
- NG_HOOK_UNREF
 - netgraph.h, 1134
- NG_HOOKLEN

- ng_message.h, 1413
- NG_HOOKSIZ
 - ng_message.h, 1413
- ng_hub.c
 - NETGRAPH_INIT, 1333
 - ng_hub_constructor, 1333
 - ng_hub_disconnect, 1333
 - ng_hub_rcvdata, 1333
 - ng_hub_typestruct, 1333
- ng_hub.h
 - NG_HUB_NODE_TYPE, 1335
 - NGM_HUB_COOKIE, 1335
- ng_hub_constructor
 - ng_hub.c, 1333
- ng_hub_disconnect
 - ng_hub.c, 1333
- NG_HUB_NODE_TYPE
 - ng_hub.h, 1335
- ng_hub_rcvdata
 - ng_hub.c, 1333
- ng_hub_typestruct
 - ng_hub.c, 1333
- ng_ID2noderef
 - ng_base.c, 1190
- NG_ID_HASH_SIZE
 - ng_base.c, 1180
- ng_ID_t
 - ng_message.h, 1413
- NG_IDHASH_FIND
 - ng_base.c, 1180
- NG_IDHASH_FN
 - ng_base.c, 1180
- ng_iface.c
 - get_hook_from_iffam, 1339
 - get_iffam_from_af, 1339
 - get_iffam_from_hook, 1339
 - get_iffam_from_name, 1339
 - gFamilies, 1342
 - iffam_p, 1339
 - M_NETGRAPH_IFACE, 1338
 - NETGRAPH_INIT, 1339
 - ng_cisco_ipaddr_type, 1342
 - ng_cisco_ipaddr_type_fields, 1343
 - ng_iface_bpftap, 1339
 - ng_iface_cmds, 1343
 - ng_iface_constructor, 1339, 1343
 - ng_iface_disconnect, 1340, 1343
 - ng_iface_ioctl, 1340
 - ng_iface_mod_event, 1340
 - ng_iface_newhook, 1340, 1343
 - ng_iface_output, 1341
 - ng_iface_rcvdata, 1341, 1343
 - ng_iface_rcvmsg, 1341, 1343
 - ng_iface_send, 1341
 - ng_iface_shutdown, 1342, 1343
 - ng_iface_start, 1342
 - ng_iface_unit, 1343
 - NUM_FAMILIES, 1338
 - priv_p, 1339
 - typestruct, 1343
- ng_iface.h
 - NGM_IFACE_BROADCAST, 1346
 - NGM_IFACE_GET_IFINDEX, 1346
 - NGM_IFACE_GET_IFNAME, 1346
 - NGM_IFACE_POINT2POINT, 1346
- ng_iface.h
 - NG_IFACE_HOOK_ATALK, 1345
 - NG_IFACE_HOOK_ATM, 1345
 - NG_IFACE_HOOK_INET, 1345
 - NG_IFACE_HOOK_INET6, 1345
 - NG_IFACE_HOOK_IPX, 1345
 - NG_IFACE_HOOK_NATM, 1345
 - NG_IFACE_IFACE_NAME, 1346
 - NG_IFACE_MTU_DEFAULT, 1346
 - NG_IFACE_MTU_MAX, 1346
 - NG_IFACE_MTU_MIN, 1346
 - NG_IFACE_NODE_TYPE, 1346
 - NGM_IFACE_COOKIE, 1346
- ng_iface_bpftap
 - ng_iface.c, 1339
- ng_iface_cmds
 - ng_iface.c, 1343
- ng_iface_constructor
 - ng_iface.c, 1339, 1343
- ng_iface_disconnect
 - ng_iface.c, 1340, 1343
- NG_IFACE_HOOK_ATALK
 - ng_iface.h, 1345
- NG_IFACE_HOOK_ATM
 - ng_iface.h, 1345
- NG_IFACE_HOOK_INET
 - ng_iface.h, 1345
- NG_IFACE_HOOK_INET6
 - ng_iface.h, 1345
- NG_IFACE_HOOK_IPX
 - ng_iface.h, 1345
- NG_IFACE_HOOK_NATM
 - ng_iface.h, 1345
- NG_IFACE_IFACE_NAME
 - ng_iface.h, 1346
- ng_iface_ioctl
 - ng_iface.c, 1340
- ng_iface_mod_event
 - ng_iface.c, 1340
- NG_IFACE_MTU_DEFAULT
 - ng_iface.h, 1346
- NG_IFACE_MTU_MAX
 - ng_iface.h, 1346

- NG_IFACE_MTU_MIN
 - ng_iface.h, 1346
- ng_iface_newhook
 - ng_iface.c, 1340, 1343
- NG_IFACE_NODE_TYPE
 - ng_iface.h, 1346
- ng_iface_output
 - ng_iface.c, 1341
- ng_iface_private, 249
 - hooks, 249
 - ifp, 249
 - node, 249
 - unit, 250
- ng_iface_rcvdata
 - ng_iface.c, 1341, 1343
- ng_iface_rcvmsg
 - ng_iface.c, 1341, 1343
- ng_iface_send
 - ng_iface.c, 1341
- ng_iface_shutdown
 - ng_iface.c, 1342, 1343
- ng_iface_start
 - ng_iface.c, 1342
- ng_iface_unit
 - ng_iface.c, 1343
- ng_int16_getAlign
 - ng_parse.c, 1458
- ng_int16_getDefault
 - ng_parse.c, 1458
- ng_int16_parse
 - ng_parse.c, 1458
- ng_int16_unparse
 - ng_parse.c, 1458
- ng_int32_getAlign
 - ng_parse.c, 1459
- ng_int32_getDefault
 - ng_parse.c, 1459
- ng_int32_parse
 - ng_parse.c, 1459
- ng_int32_unparse
 - ng_parse.c, 1459
- ng_int64_getAlign
 - ng_parse.c, 1459
- ng_int64_getDefault
 - ng_parse.c, 1459
- ng_int64_parse
 - ng_parse.c, 1459
- ng_int64_unparse
 - ng_parse.c, 1460
- ng_int8_getAlign
 - ng_parse.c, 1460
- ng_int8_getDefault
 - ng_parse.c, 1460
- ng_int8_parse
 - ng_parse.c, 1460
- ng_int8_unparse
 - ng_parse.c, 1460
- NG_INVALID
 - netgraph.h, 1134
- ng_ip_input.c
 - NETGRAPH_INIT, 1348
 - ngipi_cons, 1348, 1349
 - ngipi_disconnect, 1348, 1349
 - ngipi_rcvdata, 1349
 - typestruct, 1349
- ng_ip_input.h
 - NG_IP_INPUT_NODE_TYPE, 1350
 - NGM_IP_INPUT_COOKIE, 1350
- NG_IP_INPUT_NODE_TYPE
 - ng_ip_input.h, 1350
- ng_ipaddr_getDefault
 - ng_parse.c, 1460
- ng_ipaddr_parse
 - ng_parse.c, 1461
- ng_ipaddr_unparse
 - ng_parse.c, 1461
- ng_ipfw.c
 - fw_node, 1355
 - hpriv_p, 1353
 - MODULE_DEPEND, 1353
 - NETGRAPH_INIT, 1353
 - ng_ipfw_connect, 1353, 1355
 - ng_ipfw_constructor, 1353, 1355
 - ng_ipfw_disconnect, 1353, 1355
 - ng_ipfw_findhook, 1353, 1355
 - ng_ipfw_findhook1, 1354
 - ng_ipfw_input, 1354
 - ng_ipfw_mod_event, 1354
 - ng_ipfw_newhook, 1354, 1355
 - ng_ipfw_rcvdata, 1355
 - ng_ipfw_shutdown, 1355, 1356
 - ng_ipfw_typestruct, 1356
- ng_ipfw.h
 - NG_IPFW_IN, 1357
 - ng_ipfw_input_p, 1358
 - ng_ipfw_input_t, 1358
 - NG_IPFW_LOADED, 1357
 - NG_IPFW_NODE_TYPE, 1357
 - NG_IPFW_OUT, 1357
 - NGM_IPFW_COOKIE, 1357
 - TAGSIZ, 1358
- ng_ipfw_connect
 - ng_ipfw.c, 1353, 1355
- ng_ipfw_constructor
 - ng_ipfw.c, 1353, 1355
- ng_ipfw_disconnect
 - ng_ipfw.c, 1353, 1355
- ng_ipfw_findhook

- ng_ipfw.c, 1353, 1355
- ng_ipfw_findhook1
 - ng_ipfw.c, 1354
- ng_ipfw_hook_priv, 251
 - hook, 251
 - rulenum, 251
- NG_IPFW_IN
 - ng_ipfw.h, 1357
- ng_ipfw_input
 - ng_ipfw.c, 1354
- ng_ipfw_input_p
 - ng_ipfw.h, 1358
- ng_ipfw_input_t
 - ng_ipfw.h, 1358
- NG_IPFW_LOADED
 - ng_ipfw.h, 1357
- ng_ipfw_mod_event
 - ng_ipfw.c, 1354
- ng_ipfw_newhook
 - ng_ipfw.c, 1354, 1355
- NG_IPFW_NODE_TYPE
 - ng_ipfw.h, 1357
- NG_IPFW_OUT
 - ng_ipfw.h, 1357
- ng_ipfw_rcvdata
 - ng_ipfw.c, 1355
- ng_ipfw_shutdown
 - ng_ipfw.c, 1355, 1356
- ng_ipfw_tag, 252
 - dir, 252
 - ifp, 252
 - mt, 252
 - rule, 252
- ng_ipfw_tpestruct
 - ng_ipfw.c, 1356
- ng_item, 253
 - apply, 254
 - body, 254
 - context, 254
 - da_m, 254
 - el_dest, 254
 - el_flags, 254
 - el_hook, 254
 - el_next, 254
 - fn, 254
 - fn_arg1, 254
 - fn_arg2, 254
 - fn_fn, 255
 - msg, 255
 - msg_msg, 255
 - msg_retaddr, 255
- ng_item_fn
 - netgraph.h, 1150
- ng_ksocket.c
 - ERRROUT, 1362
 - KSF_ACCEPTING, 1362
 - KSF_CLONED, 1362
 - KSF_CONNECTING, 1362
 - KSF_EMBRYONIC, 1362
 - KSF_EOFSEEN, 1362
 - M_NETGRAPH_KSOCKET, 1362
 - NETGRAPH_INIT, 1363
 - ng_ksocket_accept_type, 1367
 - ng_ksocket_accept_type_fields, 1367
 - ng_ksocket_check_accept, 1363
 - ng_ksocket_cmds, 1367
 - ng_ksocket_connect, 1363, 1367
 - ng_ksocket_constructor, 1363, 1368
 - ng_ksocket_disconnect, 1363, 1368
 - ng_ksocket_families, 1368
 - ng_ksocket_finish_accept, 1364
 - ng_ksocket_generic_sockaddr_type, 1368
 - ng_ksocket_generic_sockdata_type, 1368
 - ng_ksocket_incoming, 1364
 - ng_ksocket_incoming2, 1364
 - ng_ksocket_newhook, 1365, 1368
 - ng_ksocket_parse, 1365
 - ng_ksocket_protos, 1369
 - ng_ksocket_rcvdata, 1365, 1369
 - ng_ksocket_rcvmsg, 1365, 1369
 - ng_ksocket_shutdown, 1366, 1369
 - ng_ksocket_sockaddr_parse, 1366
 - ng_ksocket_sockaddr_type, 1369
 - ng_ksocket_sockaddr_unparse, 1366
 - ng_ksocket_sockopt_type, 1369
 - ng_ksocket_sockopt_type_fields, 1370
 - ng_ksocket_sockoptval_type, 1370
 - ng_ksocket_types, 1370
 - ng_ksocket_tpestruct, 1370
 - ng_parse_generic_sockaddr_type_fields, 1371
 - ng_parse_generic_sockdata_getLength, 1367
 - ng_parse_sockoptval_getLength, 1367
 - OFFSETOF, 1362
 - priv_p, 1363
 - SADATA_OFFSET, 1362
- ng_ksocket.h
 - NGM_KSOCKET_ACCEPT, 1373
 - NGM_KSOCKET_BIND, 1373
 - NGM_KSOCKET_CONNECT, 1373
 - NGM_KSOCKET_GETNAME, 1373
 - NGM_KSOCKET_GETOPT, 1373
 - NGM_KSOCKET_GETPEERNAME, 1373
 - NGM_KSOCKET_LISTEN, 1373
 - NGM_KSOCKET_SETOPT, 1373
- ng_ksocket.h
 - NG_KSOCKET_MAX_OPTLEN, 1372
 - NG_KSOCKET_NODE_TYPE, 1372
 - NG_KSOCKET_SOCKOPT_INFO, 1372

- NG_KSOCKET_TAG_SOCKADDR, 1373
- NGM_KSOCKET_ACCEPT_INFO, 1373
- NGM_KSOCKET_COOKIE, 1373
- ng_ksocket_accept, 256
 - addr, 256
 - nodeid, 256
- ng_ksocket_accept_type
 - ng_ksocket.c, 1367
- ng_ksocket_accept_type_fields
 - ng_ksocket.c, 1367
- ng_ksocket_alias, 257
 - family, 257
 - name, 257
 - value, 257
- ng_ksocket_check_accept
 - ng_ksocket.c, 1363
- ng_ksocket_cmds
 - ng_ksocket.c, 1367
- ng_ksocket_connect
 - ng_ksocket.c, 1363, 1367
- ng_ksocket_constructor
 - ng_ksocket.c, 1363, 1368
- ng_ksocket_disconnect
 - ng_ksocket.c, 1363, 1368
- ng_ksocket_families
 - ng_ksocket.c, 1368
- ng_ksocket_finish_accept
 - ng_ksocket.c, 1364
- ng_ksocket_generic_sockaddr_type
 - ng_ksocket.c, 1368
- ng_ksocket_generic_sockdata_type
 - ng_ksocket.c, 1368
- ng_ksocket_incoming
 - ng_ksocket.c, 1364
- ng_ksocket_incoming2
 - ng_ksocket.c, 1364
- NG_KSOCKET_MAX_OPTLEN
 - ng_ksocket.h, 1372
- ng_ksocket_newhook
 - ng_ksocket.c, 1365, 1368
- NG_KSOCKET_NODE_TYPE
 - ng_ksocket.h, 1372
- ng_ksocket_parse
 - ng_ksocket.c, 1365
- ng_ksocket_private, 258
 - hook, 258
 - node, 258
 - so, 259
- ng_ksocket_protos
 - ng_ksocket.c, 1369
- ng_ksocket_rcvdata
 - ng_ksocket.c, 1365, 1369
- ng_ksocket_rcvmsg
 - ng_ksocket.c, 1365, 1369
- ng_ksocket_shutdown
 - ng_ksocket.c, 1366, 1369
- ng_ksocket_sockaddr_parse
 - ng_ksocket.c, 1366
- ng_ksocket_sockaddr_type
 - ng_ksocket.c, 1369
- ng_ksocket_sockaddr_unparse
 - ng_ksocket.c, 1366
- ng_ksocket_sockopt, 260
 - level, 260
 - name, 260
 - value, 260
- NG_KSOCKET_SOCKOPT_INFO
 - ng_ksocket.h, 1372
- ng_ksocket_sockopt_type
 - ng_ksocket.c, 1369
- ng_ksocket_sockopt_type_fields
 - ng_ksocket.c, 1370
- ng_ksocket_sockoptval_type
 - ng_ksocket.c, 1370
- NG_KSOCKET_TAG_SOCKADDR
 - ng_ksocket.h, 1373
- ng_ksocket_types
 - ng_ksocket.c, 1370
- ng_ksocket_typestruct
 - ng_ksocket.c, 1370
- ng_l2cap, 261
 - bdaddr, 262
 - cid, 262
 - ctl, 262
 - debug, 262
 - discon_timo, 262
 - flags, 262
 - hci, 262
 - l2c, 263
 - LIST_HEAD, 262
 - node, 263
 - num_pkts, 263
 - pkt_size, 263
- ng_l2cap.h
 - NG_L2CAP_ALERT_LEVEL, 904
 - NG_L2CAP_AUTH_PENDING, 904
 - NG_L2CAP_AUTZ_PENDING, 904
 - ng_l2cap_cfg_opt_p, 917
 - ng_l2cap_cfg_opt_val_p, 917
 - NG_L2CAP_CFG_REQ, 904
 - NG_L2CAP_CFG_RSP, 904
 - NG_L2CAP_CLOSED, 904
 - NG_L2CAP_CLT_CID, 904
 - NG_L2CAP_CLT_MTU_MAXIMUM, 905
 - NG_L2CAP_CLT_RFCOMM_DISABLED, 905
 - NG_L2CAP_CLT_SDP_DISABLED, 905
 - NG_L2CAP_CLT_TCP_DISABLED, 905

- NG_L2CAP_CMD_REJ, 905
- ng_l2cap_cmd_rej_data_p, 918
- NG_L2CAP_CON_AUTO_DISCON_TIMO, 905
- NG_L2CAP_CON_CLOSED, 905
- NG_L2CAP_CON_DYING, 905
- NG_L2CAP_CON_LP_TIMO, 906
- NG_L2CAP_CON_OPEN, 906
- NG_L2CAP_CON_OUTGOING, 906
- NG_L2CAP_CON_REQ, 906
- NG_L2CAP_CON_RSP, 906
- NG_L2CAP_CON_RX, 906
- NG_L2CAP_CON_TX, 906
- NG_L2CAP_CONFIG, 906
- NG_L2CAP_CONNLESS_MTU, 907
- NG_L2CAP_DISCON_REQ, 907
- NG_L2CAP_DISCON_RSP, 907
- ng_l2cap_discon_rsp_cp, 918
- NG_L2CAP_ECHO_REQ, 907
- NG_L2CAP_ECHO_RSP, 907
- NG_L2CAP_ERR_LEVEL, 907
- NG_L2CAP_FIRST_CID, 907
- ng_l2cap_flow_p, 918
- NG_L2CAP_FLUSH_TIMO_DEFAULT, 907
- NG_L2CAP_HOOK_CTL, 908
- NG_L2CAP_HOOK_HCI, 908
- NG_L2CAP_HOOK_L2C, 908
- NG_L2CAP_INFO_LEVEL, 908
- NG_L2CAP_INFO_REQ, 908
- NG_L2CAP_INFO_RSP, 908
- ng_l2cap_info_rsp_data_p, 918
- ng_l2cap_l2ca_discon_ind_ip, 918
- ng_l2cap_l2ca_grp_rem_member_ip, 918
- NG_L2CAP_LAST_CID, 908
- NG_L2CAP_LINK_TIMO_DEFAULT, 908
- NG_L2CAP_MAX_CHAN_NUM, 908
- NG_L2CAP_MAX_CON_NUM, 909
- NG_L2CAP_MAX_ECHO_SIZE, 909
- NG_L2CAP_MTU_DEFAULT, 909
- NG_L2CAP_MTU_MAXIMUM, 909
- NG_L2CAP_MTU_MINIMUM, 909
- NG_L2CAP_NO_INFO, 909
- NG_L2CAP_NO_RESOURCES, 909
- ng_l2cap_node_auto_discon_ep, 918
- ng_l2cap_node_debug_ep, 918
- ng_l2cap_node_flags_ep, 918
- NG_L2CAP_NODE_TYPE, 909
- NG_L2CAP_NOT_SUPPORTED, 909
- NG_L2CAP_NULL_CID, 910
- NG_L2CAP_OPEN, 910
- NG_L2CAP_OPT_CFLAG, 910
- NG_L2CAP_OPT_CFLAG_BIT, 910
- NG_L2CAP_OPT_FLUSH_TIMO, 910
- NG_L2CAP_OPT_FLUSH_TIMO_SIZE, 910
- NG_L2CAP_OPT_HINT, 910
- NG_L2CAP_OPT_HINT_BIT, 910
- NG_L2CAP_OPT_HINT_MASK, 910
- NG_L2CAP_OPT_MTU, 911
- NG_L2CAP_OPT_MTU_SIZE, 911
- NG_L2CAP_OPT_QOS, 911
- NG_L2CAP_OPT_QOS_SIZE, 911
- NG_L2CAP_PENDING, 911
- NG_L2CAP_PSM_ANY, 911
- NG_L2CAP_PSM_NOT_SUPPORTED, 911
- NG_L2CAP_PSM_RFCOMM, 911
- NG_L2CAP_PSM_SDP, 912
- NG_L2CAP_PSM_TCP, 912
- NG_L2CAP_REJ_INVALID_CID, 912
- NG_L2CAP_REJ_MTU_EXCEEDED, 912
- NG_L2CAP_REJ_NOT_UNDERSTOOD, 912
- NG_L2CAP_REJECT, 912
- NG_L2CAP_SEQUIRY_BLOCK, 912
- NG_L2CAP_SIGNAL_CID, 912
- NG_L2CAP_SUCCESS, 912
- NG_L2CAP_TIMEOUT, 913
- NG_L2CAP_UNACCEPTABLE_PARAMS, 913
- NG_L2CAP_UNKNOWN, 913
- NG_L2CAP_UNKNOWN_OPTION, 913
- NG_L2CAP_W4_L2CA_CON_RSP, 913
- NG_L2CAP_W4_L2CA_DISCON_RSP, 913
- NG_L2CAP_W4_L2CAP_CON_RSP, 913
- NG_L2CAP_W4_L2CAP_DISCON_RSP, 913
- NG_L2CAP_W4_LP_CON_CFM, 914
- NG_L2CAP_WARN_LEVEL, 914
- NGM_L2CAP_COOKIE, 914
- NGM_L2CAP_L2CA_CFG, 914
- NGM_L2CAP_L2CA_CFG_IND, 914
- NGM_L2CAP_L2CA_CFG_RSP, 914
- NGM_L2CAP_L2CA_CON, 914
- NGM_L2CAP_L2CA_CON_IND, 915
- NGM_L2CAP_L2CA_CON_RSP, 915
- NGM_L2CAP_L2CA_DISCON, 915
- NGM_L2CAP_L2CA_DISCON_IND, 915
- NGM_L2CAP_L2CA_ENABLE_CLT, 915
- NGM_L2CAP_L2CA_GET_INFO, 915
- NGM_L2CAP_L2CA_GRP_ADD_MEMBER, 915
- NGM_L2CAP_L2CA_GRP_CLOSE, 915
- NGM_L2CAP_L2CA_GRP_CREATE, 916
- NGM_L2CAP_L2CA_GRP_MEMBERSHIP, 916
- NGM_L2CAP_L2CA_GRP_REMOVE_MEMBER, 916
- NGM_L2CAP_L2CA_PING, 916
- NGM_L2CAP_L2CA_QOS_IND, 916

- NGM_L2CAP_L2CA_WRITE, 916
- NGM_L2CAP_NODE_GET_AUTO_-DISCON_TIMO, 916
- NGM_L2CAP_NODE_GET_CHAN_LIST, 916
- NGM_L2CAP_NODE_GET_CON_LIST, 917
- NGM_L2CAP_NODE_GET_DEBUG, 917
- NGM_L2CAP_NODE_GET_FLAGS, 917
- NGM_L2CAP_NODE_HOOK_INFO, 917
- NGM_L2CAP_NODE_SET_AUTO_-DISCON_TIMO, 917
- NGM_L2CAP_NODE_SET_DEBUG, 917
- NG_L2CAP_ALERT
 - ng_l2cap_var.h, 1007
- NG_L2CAP_ALERT_LEVEL
 - ng_l2cap.h, 904
- NG_L2CAP_AUTH_PENDING
 - ng_l2cap.h, 904
- NG_L2CAP_AUTZ_PENDING
 - ng_l2cap.h, 904
- NG_L2CAP_CFG_BOTH
 - ng_l2cap_var.h, 1007
- NG_L2CAP_CFG_IN
 - ng_l2cap_var.h, 1007
- ng_l2cap_cfg_opt_p
 - ng_l2cap.h, 917
- ng_l2cap_cfg_opt_val_p
 - ng_l2cap.h, 917
- ng_l2cap_cfg_opt_val_t, 264
 - flow, 264
 - flush_timo, 264
 - mtu, 264
- NG_L2CAP_CFG_OUT
 - ng_l2cap_var.h, 1007
- NG_L2CAP_CFG_REQ
 - ng_l2cap.h, 904
- NG_L2CAP_CFG_RSP
 - ng_l2cap.h, 904
- ng_l2cap_chan, 265
 - cfg_state, 266
 - con, 266
 - dcid, 266
 - flush_timo, 266
 - ident, 266
 - iflow, 266
 - imtu, 267
 - link_timo, 267
 - LIST_ENTRY, 266
 - oflow, 267
 - omtu, 267
 - psm, 267
 - scid, 267
 - state, 267
- ng_l2cap_chan_by_scid
 - ng_l2cap_misc.c, 970
 - ng_l2cap_misc.h, 979
- ng_l2cap_chan_p
 - ng_l2cap_var.h, 1009
- ng_l2cap_chan_t
 - ng_l2cap_var.h, 1009
- ng_l2cap_cleanup
 - ng_l2cap_main.c, 961
- NG_L2CAP_CLOSED
 - ng_l2cap.h, 904
- NG_L2CAP_CLT_CID
 - ng_l2cap.h, 904
- NG_L2CAP_CLT_MTU_MAXIMUM
 - ng_l2cap.h, 905
- NG_L2CAP_CLT_RFCOMM_DISABLED
 - ng_l2cap.h, 905
- NG_L2CAP_CLT_SDP_DISABLED
 - ng_l2cap.h, 905
- NG_L2CAP_CLT_TCP_DISABLED
 - ng_l2cap.h, 905
- ng_l2cap_cmd, 269
 - aux, 270
 - ch, 270
 - code, 270
 - con, 270
 - flags, 270
 - ident, 270
 - TAILQ_ENTRY, 270
 - timo, 270
 - token, 270
- ng_l2cap_cmd_by_ident
 - ng_l2cap_misc.c, 970
 - ng_l2cap_misc.h, 979
- ng_l2cap_cmd_p
 - ng_l2cap_var.h, 1009
- NG_L2CAP_CMD_PENDING
 - ng_l2cap_var.h, 1007
- NG_L2CAP_CMD_REJ
 - ng_l2cap.h, 905
- ng_l2cap_cmd_rej_data_p
 - ng_l2cap.h, 918
- ng_l2cap_cmd_rej_data_t, 271
 - cid, 271
 - dcid, 271
 - mtu, 271
 - scid, 271
- ng_l2cap_cmd_t
 - ng_l2cap_var.h, 1009
- ng_l2cap_cmdlist
 - ng_l2cap_prse.h, 985
- ng_l2cap_cmds.c
 - ng_l2cap_con_fail, 923
 - ng_l2cap_con_wakeup, 924

- ng_l2cap_process_command_timeout, 925
- ng_l2cap_cmds.h
 - _ng_l2cap_build_cfg_options, 927
 - _ng_l2cap_cfg_req, 927
 - _ng_l2cap_cfg_rsp, 927
 - _ng_l2cap_cmd_rej, 928
 - _ng_l2cap_con_req, 928
 - _ng_l2cap_con_rsp, 928
 - _ng_l2cap_discon_req, 929
 - _ng_l2cap_discon_rsp, 929
 - _ng_l2cap_echo_req, 930
 - _ng_l2cap_info_req, 930
 - _ng_l2cap_info_rsp, 931
 - ng_l2cap_con_fail, 931
 - ng_l2cap_con_wakeup, 931
 - ng_l2cap_process_command_timeout, 932
- ng_l2cap_command_timeout
 - ng_l2cap_misc.c, 970
 - ng_l2cap_misc.h, 979
- ng_l2cap_command_untimeout
 - ng_l2cap_misc.c, 970
 - ng_l2cap_misc.h, 979
- ng_l2cap_con, 272
 - con_handle, 273
 - con_timo, 273
 - flags, 273
 - ident, 273
 - l2cap, 273
 - LIST_ENTRY, 273
 - pending, 274
 - refcnt, 274
 - remote, 274
 - rx_pkt, 274
 - rx_pkt_len, 274
 - state, 274
 - TAILQ_HEAD, 273
 - tx_pkt, 275
- NG_L2CAP_CON_AUTO_DISCON_TIMO
 - ng_l2cap.h, 905
- ng_l2cap_con_by_addr
 - ng_l2cap_misc.c, 971
 - ng_l2cap_misc.h, 980
- ng_l2cap_con_by_handle
 - ng_l2cap_misc.c, 971
 - ng_l2cap_misc.h, 980
- NG_L2CAP_CON_CLOSED
 - ng_l2cap.h, 905
- NG_L2CAP_CON_DYING
 - ng_l2cap.h, 905
- ng_l2cap_con_fail
 - ng_l2cap_cmds.c, 923
 - ng_l2cap_cmds.h, 931
- NG_L2CAP_CON_LP_TIMO
 - ng_l2cap.h, 906
- NG_L2CAP_CON_OPEN
 - ng_l2cap.h, 906
- NG_L2CAP_CON_OUTGOING
 - ng_l2cap.h, 906
- ng_l2cap_con_p
 - ng_l2cap_var.h, 1009
- ng_l2cap_con_ref
 - ng_l2cap_misc.c, 971
 - ng_l2cap_misc.h, 980
- NG_L2CAP_CON_REQ
 - ng_l2cap.h, 906
- NG_L2CAP_CON_RSP
 - ng_l2cap.h, 906
- NG_L2CAP_CON_RX
 - ng_l2cap.h, 906
- ng_l2cap_con_t
 - ng_l2cap_var.h, 1009
- NG_L2CAP_CON_TX
 - ng_l2cap.h, 906
- ng_l2cap_con_unref
 - ng_l2cap_misc.c, 971
 - ng_l2cap_misc.h, 980
- ng_l2cap_con_wakeup
 - ng_l2cap_cmds.c, 924
 - ng_l2cap_cmds.h, 931
- NG_L2CAP_CONFIG
 - ng_l2cap.h, 906
- ng_l2cap_connect
 - ng_l2cap_main.c, 961, 966
- NG_L2CAP_CONNLESS_MTU
 - ng_l2cap.h, 907
- ng_l2cap_constructor
 - ng_l2cap_main.c, 962, 966
- ng_l2cap_default_flow
 - ng_l2cap_misc.c, 972
 - ng_l2cap_misc.h, 981
- ng_l2cap_default_rcvmsg
 - ng_l2cap_main.c, 962, 966
- ng_l2cap_destroy_channels
 - ng_l2cap_main.c, 962
- NG_L2CAP_DISCON_REQ
 - ng_l2cap.h, 907
- NG_L2CAP_DISCON_RSP
 - ng_l2cap.h, 907
- ng_l2cap_discon_rsp_cp
 - ng_l2cap.h, 918
- ng_l2cap_discon_timeout
 - ng_l2cap_misc.c, 972
 - ng_l2cap_misc.h, 981
- ng_l2cap_discon_untimeout
 - ng_l2cap_misc.c, 972
 - ng_l2cap_misc.h, 981
- ng_l2cap_disconnect
 - ng_l2cap_main.c, 963, 966

- NG_L2CAP_ECHO_REQ
 - ng_l2cap.h, 907
- NG_L2CAP_ECHO_RSP
 - ng_l2cap.h, 907
- NG_L2CAP_ERR
 - ng_l2cap_var.h, 1007
- NG_L2CAP_ERR_LEVEL
 - ng_l2cap.h, 907
- ng_l2cap_evnt.c
 - get_next_l2cap_opt, 936
 - ng_l2cap_process_cfg_req, 936
 - ng_l2cap_process_cfg_rsp, 936
 - ng_l2cap_process_cmd_rej, 936
 - ng_l2cap_process_con_req, 937
 - ng_l2cap_process_con_rsp, 938
 - ng_l2cap_process_discon_req, 938
 - ng_l2cap_process_discon_rsp, 938
 - ng_l2cap_process_echo_req, 939
 - ng_l2cap_process_echo_rsp, 939
 - ng_l2cap_process_info_req, 940
 - ng_l2cap_process_info_rsp, 940
 - ng_l2cap_process_signal_cmd, 940
 - ng_l2cap_receive, 941
 - send_l2cap_cfg_rsp, 942
 - send_l2cap_con_rej, 942
 - send_l2cap_reject, 943
- ng_l2cap_evnt.h
 - ng_l2cap_receive, 944
- NG_L2CAP_FIRST_CID
 - ng_l2cap.h, 907
- NG_L2CAP_FIRST_IDENT
 - ng_l2cap_var.h, 1008
- ng_l2cap_flow_p
 - ng_l2cap.h, 918
- NG_L2CAP_FLUSH_TIMO_DEFAULT
 - ng_l2cap.h, 907
- ng_l2cap_free_chan
 - ng_l2cap_misc.c, 972
 - ng_l2cap_misc.h, 981
- ng_l2cap_free_cmd
 - ng_l2cap_misc.h, 978
- ng_l2cap_free_con
 - ng_l2cap_misc.c, 973
 - ng_l2cap_misc.h, 982
- ng_l2cap_get_cid
 - ng_l2cap_misc.c, 973
- ng_l2cap_get_ident
 - ng_l2cap_misc.c, 973
 - ng_l2cap_misc.h, 982
- NG_L2CAP_HOOK_CTL
 - ng_l2cap.h, 908
- NG_L2CAP_HOOK_HCI
 - ng_l2cap.h, 908
- NG_L2CAP_HOOK_L2C
 - ng_l2cap.h, 908
- NG_L2CAP_INFO
 - ng_l2cap_var.h, 1008
- NG_L2CAP_INFO_LEVEL
 - ng_l2cap.h, 908
- NG_L2CAP_INFO_REQ
 - ng_l2cap.h, 908
- NG_L2CAP_INFO_RSP
 - ng_l2cap.h, 908
- ng_l2cap_info_rsp_data_p
 - ng_l2cap.h, 918
- ng_l2cap_info_rsp_data_t, 276
 - mtu, 276
- ng_l2cap_l2ca_cfg_ind
 - ng_l2cap_ulpi.c, 988
 - ng_l2cap_ulpi.h, 998
- ng_l2cap_l2ca_cfg_ind_ip, 277
 - flush_timo, 277
 - iflow, 277
 - lcid, 277
 - omtu, 277
- ng_l2cap_l2ca_cfg_ip, 278
 - flush_timo, 278
 - imtu, 278
 - lcid, 278
 - link_timo, 278
 - oflow, 278
- ng_l2cap_l2ca_cfg_op, 279
 - flush_timo, 279
 - imtu, 279
 - oflow, 279
 - result, 279
- ng_l2cap_l2ca_cfg_req
 - ng_l2cap_ulpi.c, 988
 - ng_l2cap_ulpi.h, 998
- ng_l2cap_l2ca_cfg_rsp
 - ng_l2cap_ulpi.c, 988
 - ng_l2cap_ulpi.h, 998
- ng_l2cap_l2ca_cfg_rsp_ip, 280
 - iflow, 280
 - lcid, 280
 - omtu, 280
- ng_l2cap_l2ca_cfg_rsp_op, 281
 - result, 281
- ng_l2cap_l2ca_cfg_rsp_req
 - ng_l2cap_ulpi.c, 989
 - ng_l2cap_ulpi.h, 998
- ng_l2cap_l2ca_cfg_rsp_rsp
 - ng_l2cap_ulpi.c, 989
 - ng_l2cap_ulpi.h, 999
- ng_l2cap_l2ca_clt_receive
 - ng_l2cap_ulpi.c, 989
 - ng_l2cap_ulpi.h, 999
- ng_l2cap_l2ca_con_ind

- ng_l2cap_ulpi.c, 990
- ng_l2cap_ulpi.h, 999
- ng_l2cap_l2ca_con_ind_ip, 282
 - bdaddr, 282
 - ident, 282
 - lcid, 282
 - psm, 282
 - unused, 282
- ng_l2cap_l2ca_con_ip, 283
 - bdaddr, 283
 - psm, 283
- ng_l2cap_l2ca_con_op, 284
 - lcid, 284
 - result, 284
 - status, 284
- ng_l2cap_l2ca_con_req
 - ng_l2cap_ulpi.c, 990
 - ng_l2cap_ulpi.h, 999
- ng_l2cap_l2ca_con_rsp
 - ng_l2cap_ulpi.c, 990
 - ng_l2cap_ulpi.h, 1000
- ng_l2cap_l2ca_con_rsp_ip, 285
 - bdaddr, 285
 - ident, 285
 - lcid, 285
 - result, 285
 - status, 285
 - unused, 285
- ng_l2cap_l2ca_con_rsp_op, 286
 - result, 286
- ng_l2cap_l2ca_con_rsp_req
 - ng_l2cap_ulpi.c, 991
 - ng_l2cap_ulpi.h, 1000
- ng_l2cap_l2ca_con_rsp_rsp
 - ng_l2cap_ulpi.c, 991
 - ng_l2cap_ulpi.h, 1001
- ng_l2cap_l2ca_discon_ind
 - ng_l2cap_ulpi.c, 991
 - ng_l2cap_ulpi.h, 1001
- ng_l2cap_l2ca_discon_ind_ip
 - ng_l2cap.h, 918
- ng_l2cap_l2ca_discon_ip, 287
 - lcid, 287
- ng_l2cap_l2ca_discon_op, 288
 - result, 288
- ng_l2cap_l2ca_discon_req
 - ng_l2cap_ulpi.c, 991
 - ng_l2cap_ulpi.h, 1001
- ng_l2cap_l2ca_discon_rsp
 - ng_l2cap_ulpi.c, 992
 - ng_l2cap_ulpi.h, 1002
- ng_l2cap_l2ca_enable_clt
 - ng_l2cap_ulpi.c, 992
 - ng_l2cap_ulpi.h, 1002
- ng_l2cap_l2ca_enable_clt_ip, 289
 - enable, 289
 - psm, 289
- ng_l2cap_l2ca_get_info_ip, 290
 - bdaddr, 290
 - info_type, 290
- ng_l2cap_l2ca_get_info_op, 291
 - info_size, 291
 - result, 291
- ng_l2cap_l2ca_get_info_req
 - ng_l2cap_ulpi.c, 992
 - ng_l2cap_ulpi.h, 1002
- ng_l2cap_l2ca_get_info_rsp
 - ng_l2cap_ulpi.c, 993
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_add_member_ip, 292
 - bdaddr, 292
 - lcid, 292
- ng_l2cap_l2ca_grp_add_member_op, 293
 - result, 293
- ng_l2cap_l2ca_grp_add_member_req
 - ng_l2cap_ulpi.c, 993
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_add_member_rsp
 - ng_l2cap_ulpi.c, 993
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_close
 - ng_l2cap_ulpi.c, 993
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_close_ip, 294
 - lcid, 294
- ng_l2cap_l2ca_grp_create
 - ng_l2cap_ulpi.c, 993
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_create_ip, 295
 - psm, 295
- ng_l2cap_l2ca_grp_create_op, 296
 - lcid, 296
- ng_l2cap_l2ca_grp_get_members
 - ng_l2cap_ulpi.c, 994
 - ng_l2cap_ulpi.h, 1003
- ng_l2cap_l2ca_grp_get_members_ip, 297
 - lcid, 297
- ng_l2cap_l2ca_grp_get_members_op, 298
 - nmembers, 298
 - result, 298
- ng_l2cap_l2ca_grp_rem_member
 - ng_l2cap_ulpi.c, 994
 - ng_l2cap_ulpi.h, 1004
- ng_l2cap_l2ca_grp_rem_member_ip
 - ng_l2cap.h, 918
- ng_l2cap_l2ca_ping_ip, 299
 - bdaddr, 299
 - echo_size, 299

- ng_l2cap_l2ca_ping_op, 300
 - bdaddr, 300
 - echo_size, 300
 - result, 300
- ng_l2cap_l2ca_ping_req
 - ng_l2cap_ulpi.c, 994
 - ng_l2cap_ulpi.h, 1004
- ng_l2cap_l2ca_ping_rsp
 - ng_l2cap_ulpi.c, 994
 - ng_l2cap_ulpi.h, 1004
- ng_l2cap_l2ca_qos_ind
 - ng_l2cap_ulpi.c, 995
 - ng_l2cap_ulpi.h, 1004
- ng_l2cap_l2ca_qos_ind_ip, 301
 - bdaddr, 301
- ng_l2cap_l2ca_receive
 - ng_l2cap_ulpi.c, 995
 - ng_l2cap_ulpi.h, 1005
- ng_l2cap_l2ca_write_op, 302
 - lcid, 302
 - length, 302
 - result, 302
- ng_l2cap_l2ca_write_req
 - ng_l2cap_ulpi.c, 995
 - ng_l2cap_ulpi.h, 1005
- ng_l2cap_l2ca_write_rsp
 - ng_l2cap_ulpi.c, 995
 - ng_l2cap_ulpi.h, 1005
- NG_L2CAP_LAST_CID
 - ng_l2cap.h, 908
- NG_L2CAP_LAST_IDENT
 - ng_l2cap_var.h, 1008
- ng_l2cap_link_cmd
 - ng_l2cap_misc.h, 978
- NG_L2CAP_LINK_TIMO_DEFAULT
 - ng_l2cap.h, 908
- ng_l2cap_llpi.c
 - ng_l2cap_lp_con_cfm, 947
 - ng_l2cap_lp_con_ind, 948
 - ng_l2cap_lp_con_req, 948
 - ng_l2cap_lp_deliver, 949
 - ng_l2cap_lp_discon_ind, 950
 - ng_l2cap_lp_qos_cfm, 950
 - ng_l2cap_lp_qos_ind, 950
 - ng_l2cap_lp_qos_req, 950
 - ng_l2cap_lp_receive, 951
 - ng_l2cap_lp_send, 951
 - ng_l2cap_process_discon_timeout, 951
 - ng_l2cap_process_lp_timeout, 952
- ng_l2cap_llpi.h
 - ng_l2cap_lp_con_cfm, 953
 - ng_l2cap_lp_con_ind, 954
 - ng_l2cap_lp_con_req, 954
 - ng_l2cap_lp_deliver, 955
 - ng_l2cap_lp_discon_ind, 955
 - ng_l2cap_lp_qos_cfm, 956
 - ng_l2cap_lp_qos_ind, 956
 - ng_l2cap_lp_qos_req, 956
 - ng_l2cap_lp_receive, 957
 - ng_l2cap_lp_send, 957
 - ng_l2cap_process_discon_timeout, 957
 - ng_l2cap_process_lp_timeout, 958
- ng_l2cap_lower_rcvmsg
 - ng_l2cap_main.c, 963, 966
- ng_l2cap_lp_con_cfm
 - ng_l2cap_llpi.c, 947
 - ng_l2cap_llpi.h, 953
- ng_l2cap_lp_con_ind
 - ng_l2cap_llpi.c, 948
 - ng_l2cap_llpi.h, 954
- ng_l2cap_lp_con_req
 - ng_l2cap_llpi.c, 948
 - ng_l2cap_llpi.h, 954
- ng_l2cap_lp_deliver
 - ng_l2cap_llpi.c, 949
 - ng_l2cap_llpi.h, 955
- ng_l2cap_lp_discon_ind
 - ng_l2cap_llpi.c, 950
 - ng_l2cap_llpi.h, 955
- ng_l2cap_lp_qos_cfm
 - ng_l2cap_llpi.c, 950
 - ng_l2cap_llpi.h, 956
- ng_l2cap_lp_qos_ind
 - ng_l2cap_llpi.c, 950
 - ng_l2cap_llpi.h, 956
- ng_l2cap_lp_qos_req
 - ng_l2cap_llpi.c, 950
 - ng_l2cap_llpi.h, 956
- ng_l2cap_lp_receive
 - ng_l2cap_llpi.c, 951
 - ng_l2cap_llpi.h, 957
- ng_l2cap_lp_send
 - ng_l2cap_llpi.c, 951
 - ng_l2cap_llpi.h, 957
- ng_l2cap_lp_timeout
 - ng_l2cap_misc.c, 974
 - ng_l2cap_misc.h, 982
- ng_l2cap_lp_untimeout
 - ng_l2cap_misc.c, 974
 - ng_l2cap_misc.h, 983
- NG_L2CAP_M_PULLUP
 - ng_l2cap_var.h, 1008
- ng_l2cap_main.c
 - M_NETGRAPH_L2CAP, 961
 - MODULE_DEPEND, 961
 - MODULE_VERSION, 961
 - NETGRAPH_INIT, 961
 - ng_l2cap_cleanup, 961

- ng_l2cap_connect, 961, 966
- ng_l2cap_constructor, 962, 966
- ng_l2cap_default_rcvmsg, 962, 966
- ng_l2cap_destroy_channels, 962
- ng_l2cap_disconnect, 963, 966
- ng_l2cap_lower_rcvmsg, 963, 966
- ng_l2cap_newhook, 964, 966
- ng_l2cap_rcvdata, 964, 966
- ng_l2cap_shutdown, 964, 966
- ng_l2cap_upper_rcvmsg, 965, 966
- typestruct, 966
- NG_L2CAP_MAX_CHAN_NUM
 - ng_l2cap.h, 908
- NG_L2CAP_MAX_CON_NUM
 - ng_l2cap.h, 909
- NG_L2CAP_MAX_ECHO_SIZE
 - ng_l2cap.h, 909
- ng_l2cap_misc.c
 - ng_l2cap_chan_by_scid, 970
 - ng_l2cap_cmd_by_ident, 970
 - ng_l2cap_command_timeout, 970
 - ng_l2cap_command_untimeout, 970
 - ng_l2cap_con_by_addr, 971
 - ng_l2cap_con_by_handle, 971
 - ng_l2cap_con_ref, 971
 - ng_l2cap_con_unref, 971
 - ng_l2cap_default_flow, 972
 - ng_l2cap_discon_timeout, 972
 - ng_l2cap_discon_untimeout, 972
 - ng_l2cap_free_chan, 972
 - ng_l2cap_free_con, 973
 - ng_l2cap_get_cid, 973
 - ng_l2cap_get_ident, 973
 - ng_l2cap_lp_timeout, 974
 - ng_l2cap_lp_untimeout, 974
 - ng_l2cap_new_chan, 974
 - ng_l2cap_new_cmd, 975
 - ng_l2cap_new_con, 975
 - ng_l2cap_prepend, 975
 - ng_l2cap_send_hook_info, 975
- ng_l2cap_misc.h
 - ng_l2cap_chan_by_scid, 979
 - ng_l2cap_cmd_by_ident, 979
 - ng_l2cap_command_timeout, 979
 - ng_l2cap_command_untimeout, 979
 - ng_l2cap_con_by_addr, 980
 - ng_l2cap_con_by_handle, 980
 - ng_l2cap_con_ref, 980
 - ng_l2cap_con_unref, 980
 - ng_l2cap_default_flow, 981
 - ng_l2cap_discon_timeout, 981
 - ng_l2cap_discon_untimeout, 981
 - ng_l2cap_free_chan, 981
 - ng_l2cap_free_cmd, 978
 - ng_l2cap_free_con, 982
 - ng_l2cap_get_ident, 982
 - ng_l2cap_link_cmd, 978
 - ng_l2cap_lp_timeout, 982
 - ng_l2cap_lp_untimeout, 983
 - ng_l2cap_new_chan, 983
 - ng_l2cap_new_cmd, 983
 - ng_l2cap_new_con, 984
 - ng_l2cap_prepend, 984
 - ng_l2cap_send_hook_info, 984
 - ng_l2cap_unlink_cmd, 978
- NG_L2CAP_MTU_DEFAULT
 - ng_l2cap.h, 909
- NG_L2CAP_MTU_MAXIMUM
 - ng_l2cap.h, 909
- NG_L2CAP_MTU_MINIMUM
 - ng_l2cap.h, 909
- ng_l2cap_new_chan
 - ng_l2cap_misc.c, 974
 - ng_l2cap_misc.h, 983
- ng_l2cap_new_cmd
 - ng_l2cap_misc.c, 975
 - ng_l2cap_misc.h, 983
- ng_l2cap_new_con
 - ng_l2cap_misc.c, 975
 - ng_l2cap_misc.h, 984
- ng_l2cap_newhook
 - ng_l2cap_main.c, 964, 966
- NG_L2CAP_NO_INFO
 - ng_l2cap.h, 909
- NG_L2CAP_NO_RESOURCES
 - ng_l2cap.h, 909
- ng_l2cap_node_auto_discon_ep
 - ng_l2cap.h, 918
- ng_l2cap_node_chan_ep, 303
 - dcid, 303
 - imtu, 303
 - omtu, 303
 - psm, 303
 - remote, 303
 - scid, 303
 - state, 304
- ng_l2cap_node_chan_list_ep, 305
 - num_channels, 305
- ng_l2cap_node_con_ep, 306
 - con_handle, 306
 - flags, 306
 - pending, 306
 - remote, 306
 - state, 306
- ng_l2cap_node_con_list_ep, 307
 - num_connections, 307
- ng_l2cap_node_debug_ep
 - ng_l2cap.h, 918

- ng_l2cap_node_flags_ep
ng_l2cap.h, 918
- NG_L2CAP_NODE_TYPE
ng_l2cap.h, 909
- NG_L2CAP_NOT_SUPPORTED
ng_l2cap.h, 909
- NG_L2CAP_NULL_CID
ng_l2cap.h, 910
- NG_L2CAP_NULL_IDENT
ng_l2cap_var.h, 1008
- NG_L2CAP_OPEN
ng_l2cap.h, 910
- NG_L2CAP_OPT_CFLAG
ng_l2cap.h, 910
- NG_L2CAP_OPT_CFLAG_BIT
ng_l2cap.h, 910
- NG_L2CAP_OPT_FLUSH_TIMO
ng_l2cap.h, 910
- NG_L2CAP_OPT_FLUSH_TIMO_SIZE
ng_l2cap.h, 910
- NG_L2CAP_OPT_HINT
ng_l2cap.h, 910
- NG_L2CAP_OPT_HINT_BIT
ng_l2cap.h, 910
- NG_L2CAP_OPT_HINT_MASK
ng_l2cap.h, 910
- NG_L2CAP_OPT_MTU
ng_l2cap.h, 911
- NG_L2CAP_OPT_MTU_SIZE
ng_l2cap.h, 911
- NG_L2CAP_OPT_QOS
ng_l2cap.h, 911
- NG_L2CAP_OPT_QOS_SIZE
ng_l2cap.h, 911
- ng_l2cap_p
ng_l2cap_var.h, 1009
- NG_L2CAP_PENDING
ng_l2cap.h, 911
- ng_l2cap_prepend
ng_l2cap_misc.c, 975
ng_l2cap_misc.h, 984
- ng_l2cap_process_cfg_req
ng_l2cap_evnt.c, 936
- ng_l2cap_process_cfg_rsp
ng_l2cap_evnt.c, 936
- ng_l2cap_process_cmd_rej
ng_l2cap_evnt.c, 936
- ng_l2cap_process_command_timeout
ng_l2cap_cmds.c, 925
ng_l2cap_cmds.h, 932
- ng_l2cap_process_con_req
ng_l2cap_evnt.c, 937
- ng_l2cap_process_con_rsp
ng_l2cap_evnt.c, 938
- ng_l2cap_process_discon_req
ng_l2cap_evnt.c, 938
- ng_l2cap_process_discon_rsp
ng_l2cap_evnt.c, 938
- ng_l2cap_process_discon_timeout
ng_l2cap_llpi.c, 951
ng_l2cap_llpi.h, 957
- ng_l2cap_process_echo_req
ng_l2cap_evnt.c, 939
- ng_l2cap_process_echo_rsp
ng_l2cap_evnt.c, 939
- ng_l2cap_process_info_req
ng_l2cap_evnt.c, 940
- ng_l2cap_process_info_rsp
ng_l2cap_evnt.c, 940
- ng_l2cap_process_lp_timeout
ng_l2cap_llpi.c, 952
ng_l2cap_llpi.h, 958
- ng_l2cap_process_signal_cmd
ng_l2cap_evnt.c, 940
- ng_l2cap_prse.h
ng_l2cap_cmdlist, 985
- NG_L2CAP_PSM_ANY
ng_l2cap.h, 911
- NG_L2CAP_PSM_NOT_SUPPORTED
ng_l2cap.h, 911
- NG_L2CAP_PSM_RFCOMM
ng_l2cap.h, 911
- NG_L2CAP_PSM_SDP
ng_l2cap.h, 912
- NG_L2CAP_PSM_TCP
ng_l2cap.h, 912
- ng_l2cap_rcvdata
ng_l2cap_main.c, 964, 966
- ng_l2cap_receive
ng_l2cap_evnt.c, 941
ng_l2cap_evnt.h, 944
- NG_L2CAP_REJ_INVALID_CID
ng_l2cap.h, 912
- NG_L2CAP_REJ_MTU_EXCEEDED
ng_l2cap.h, 912
- NG_L2CAP_REJ_NOT_UNDERSTOOD
ng_l2cap.h, 912
- NG_L2CAP_REJECT
ng_l2cap.h, 912
- ng_l2cap_send_hook_info
ng_l2cap_misc.c, 975
ng_l2cap_misc.h, 984
- NG_L2CAP_SEQUIRY_BLOCK
ng_l2cap.h, 912
- ng_l2cap_shutdown
ng_l2cap_main.c, 964, 966
- NG_L2CAP_SIGNAL_CID
ng_l2cap.h, 912

- NG_L2CAP_SUCCESS
 - ng_l2cap.h, 912
- ng_l2cap_t
 - ng_l2cap_var.h, 1009
- NG_L2CAP_TIMEOUT
 - ng_l2cap.h, 913
- ng_l2cap_ulpi.c
 - ng_l2cap_l2ca_cfg_ind, 988
 - ng_l2cap_l2ca_cfg_req, 988
 - ng_l2cap_l2ca_cfg_rsp, 988
 - ng_l2cap_l2ca_cfg_rsp_req, 989
 - ng_l2cap_l2ca_cfg_rsp_rsp, 989
 - ng_l2cap_l2ca_clt_receive, 989
 - ng_l2cap_l2ca_con_ind, 990
 - ng_l2cap_l2ca_con_req, 990
 - ng_l2cap_l2ca_con_rsp, 990
 - ng_l2cap_l2ca_con_rsp_req, 991
 - ng_l2cap_l2ca_con_rsp_rsp, 991
 - ng_l2cap_l2ca_discon_ind, 991
 - ng_l2cap_l2ca_discon_req, 991
 - ng_l2cap_l2ca_discon_rsp, 992
 - ng_l2cap_l2ca_enable_clt, 992
 - ng_l2cap_l2ca_get_info_req, 992
 - ng_l2cap_l2ca_get_info_rsp, 993
 - ng_l2cap_l2ca_grp_add_member_req, 993
 - ng_l2cap_l2ca_grp_add_member_rsp, 993
 - ng_l2cap_l2ca_grp_close, 993
 - ng_l2cap_l2ca_grp_create, 993
 - ng_l2cap_l2ca_grp_get_members, 994
 - ng_l2cap_l2ca_grp_rem_member, 994
 - ng_l2cap_l2ca_ping_req, 994
 - ng_l2cap_l2ca_ping_rsp, 994
 - ng_l2cap_l2ca_qos_ind, 995
 - ng_l2cap_l2ca_receive, 995
 - ng_l2cap_l2ca_write_req, 995
 - ng_l2cap_l2ca_write_rsp, 995
- ng_l2cap_ulpi.h
 - ng_l2cap_l2ca_cfg_ind, 998
 - ng_l2cap_l2ca_cfg_req, 998
 - ng_l2cap_l2ca_cfg_rsp, 998
 - ng_l2cap_l2ca_cfg_rsp_req, 998
 - ng_l2cap_l2ca_cfg_rsp_rsp, 999
 - ng_l2cap_l2ca_clt_receive, 999
 - ng_l2cap_l2ca_con_ind, 999
 - ng_l2cap_l2ca_con_req, 999
 - ng_l2cap_l2ca_con_rsp, 1000
 - ng_l2cap_l2ca_con_rsp_req, 1000
 - ng_l2cap_l2ca_con_rsp_rsp, 1001
 - ng_l2cap_l2ca_discon_ind, 1001
 - ng_l2cap_l2ca_discon_req, 1001
 - ng_l2cap_l2ca_discon_rsp, 1002
 - ng_l2cap_l2ca_enable_clt, 1002
 - ng_l2cap_l2ca_get_info_req, 1002
 - ng_l2cap_l2ca_get_info_rsp, 1003
 - ng_l2cap_l2ca_grp_add_member_req, 1003
 - ng_l2cap_l2ca_grp_add_member_rsp, 1003
 - ng_l2cap_l2ca_grp_close, 1003
 - ng_l2cap_l2ca_grp_create, 1003
 - ng_l2cap_l2ca_grp_get_members, 1003
 - ng_l2cap_l2ca_grp_rem_member, 1004
 - ng_l2cap_l2ca_ping_req, 1004
 - ng_l2cap_l2ca_ping_rsp, 1004
 - ng_l2cap_l2ca_qos_ind, 1004
 - ng_l2cap_l2ca_receive, 1005
 - ng_l2cap_l2ca_write_req, 1005
 - ng_l2cap_l2ca_write_rsp, 1005
- NG_L2CAP_UNACCEPTABLE_PARAMS
 - ng_l2cap.h, 913
- NG_L2CAP_UNKNOWN
 - ng_l2cap.h, 913
- NG_L2CAP_UNKNOWN_OPTION
 - ng_l2cap.h, 913
- ng_l2cap_unlink_cmd
 - ng_l2cap_misc.h, 978
- ng_l2cap_upper_rcvmsg
 - ng_l2cap_main.c, 965, 966
- ng_l2cap_var.h
 - M_NETGRAPH_L2CAP, 1007
 - NG_L2CAP_ALERT, 1007
 - NG_L2CAP_CFG_BOTH, 1007
 - NG_L2CAP_CFG_IN, 1007
 - NG_L2CAP_CFG_OUT, 1007
 - ng_l2cap_chan_p, 1009
 - ng_l2cap_chan_t, 1009
 - ng_l2cap_cmd_p, 1009
 - NG_L2CAP_CMD_PENDING, 1007
 - ng_l2cap_cmd_t, 1009
 - ng_l2cap_con_p, 1009
 - ng_l2cap_con_t, 1009
 - NG_L2CAP_ERR, 1007
 - NG_L2CAP_FIRST_IDENT, 1008
 - NG_L2CAP_INFO, 1008
 - NG_L2CAP_LAST_IDENT, 1008
 - NG_L2CAP_M_PULLUP, 1008
 - NG_L2CAP_NULL_IDENT, 1008
 - ng_l2cap_p, 1009
 - ng_l2cap_t, 1009
 - NG_L2CAP_WARN, 1008
- NG_L2CAP_W4_L2CA_CON_RSP
 - ng_l2cap.h, 913
- NG_L2CAP_W4_L2CA_DISCON_RSP
 - ng_l2cap.h, 913
- NG_L2CAP_W4_L2CAP_CON_RSP
 - ng_l2cap.h, 913
- NG_L2CAP_W4_L2CAP_DISCON_RSP
 - ng_l2cap.h, 913
- NG_L2CAP_W4_LP_CON_CFM
 - ng_l2cap.h, 914

- NG_L2CAP_WARN
 - ng_l2cap_var.h, 1008
- NG_L2CAP_WARN_LEVEL
 - ng_l2cap.h, 914
- ng_l2tp.c
 - hookpriv_p, 1379
 - L2TP_CONTROL_DSEQ, 1376
 - L2TP_COPY_MBUF, 1376
 - L2TP_CTRL_0BITS, 1376
 - L2TP_CTRL_1BITS, 1377
 - L2TP_CTRL_HDR, 1377
 - L2TP_DATA_0BITS, 1377
 - L2TP_DATA_1BITS, 1377
 - L2TP_DATA_HDR, 1377
 - L2TP_DELAYED_ACK, 1377
 - L2TP_ENABLE_DSEQ, 1377
 - L2TP_HDR_CTRL, 1377
 - L2TP_HDR_LEN, 1378
 - L2TP_HDR_OFF, 1378
 - L2TP_HDR_PRIO, 1378
 - L2TP_HDR_SEQ, 1378
 - L2TP_HDR_VERS_MASK, 1378
 - L2TP_HDR_VERSION, 1378
 - L2TP_MAX_REXMIT, 1378
 - L2TP_MAX_REXMIT_TO, 1378
 - L2TP_MAX_XWIN, 1378
 - L2TP_SEQ_CHECK, 1379
 - L2TP_SEQ_DIFF, 1379
 - M_NETGRAPH_L2TP, 1379
 - memmove, 1379
 - NETGRAPH_INIT, 1379
 - ng_l2tp_cmdlist, 1385
 - ng_l2tp_config_type, 1385
 - ng_l2tp_config_type_fields, 1385
 - ng_l2tp_constructor, 1379, 1385
 - ng_l2tp_disconnect, 1380, 1386
 - ng_l2tp_find_session, 1380, 1386
 - ng_l2tp_newhook, 1380, 1386
 - ng_l2tp_rcvdata, 1380, 1386
 - ng_l2tp_rcvmsg, 1381, 1386
 - ng_l2tp_rcv_ctrl, 1381
 - ng_l2tp_rcv_data, 1381
 - ng_l2tp_rcv_lower, 1382
 - ng_l2tp_reset_session, 1382, 1386
 - ng_l2tp_seq_adjust, 1382
 - ng_l2tp_seq_config_fields, 1386
 - ng_l2tp_seq_config_type, 1386
 - ng_l2tp_seq_failure, 1383
 - ng_l2tp_seq_init, 1383
 - ng_l2tp_seq_rack_timeout, 1383
 - ng_l2tp_seq_rcv_nr, 1383
 - ng_l2tp_seq_rcv_ns, 1383
 - ng_l2tp_seq_reset, 1384
 - ng_l2tp_seq_set, 1384
 - ng_l2tp_seq_xack_timeout, 1384
 - ng_l2tp_sess_config_type, 1386
 - ng_l2tp_sess_config_type_fields, 1387
 - ng_l2tp_session_stats_type, 1387
 - ng_l2tp_session_stats_type_fields, 1387
 - ng_l2tp_shutdown, 1384, 1387
 - ng_l2tp_stats_type, 1387
 - ng_l2tp_stats_type_fields, 1387
 - ng_l2tp_tpestruct, 1387
 - ng_l2tp_xmit_ctrl, 1385
 - priv_p, 1379
- ng_l2tp.h
 - NGM_L2TP_ACK_FAILURE, 1392
 - NGM_L2TP_CLR_SESSION_STATS, 1392
 - NGM_L2TP_CLR_STATS, 1392
 - NGM_L2TP_GET_CONFIG, 1392
 - NGM_L2TP_GET_SESS_CONFIG, 1392
 - NGM_L2TP_GET_SESSION_STATS, 1392
 - NGM_L2TP_GET_STATS, 1392
 - NGM_L2TP_GETCLR_SESSION_STATS, 1392
 - NGM_L2TP_GETCLR_STATS, 1392
 - NGM_L2TP_SET_CONFIG, 1392
 - NGM_L2TP_SET_SEQ, 1392
 - NGM_L2TP_SET_SESS_CONFIG, 1392
- ng_l2tp.h
 - NG_L2TP_CONFIG_TYPE_INFO, 1389
 - NG_L2TP_HOOK_CTRL, 1390
 - NG_L2TP_HOOK_LOWER, 1390
 - NG_L2TP_HOOK_SESSION_F, 1390
 - NG_L2TP_HOOK_SESSION_P, 1390
 - NG_L2TP_NODE_TYPE, 1390
 - NG_L2TP_SEQ_CONFIG_TYPE_INFO, 1390
 - NG_L2TP_SESS_CONFIG_TYPE_INFO, 1390
 - NG_L2TP_SESSION_STATS_TYPE_INFO, 1391
 - NG_L2TP_STATS_TYPE_INFO, 1391
 - NGM_L2TP_COOKIE, 1391
- ng_l2tp_cmdlist
 - ng_l2tp.c, 1385
- ng_l2tp_config, 308
 - enabled, 308
 - match_id, 308
 - peer_id, 308
 - peer_win, 308
 - rexmit_max, 308
 - rexmit_max_to, 308
 - tunnel_id, 309
- ng_l2tp_config_type
 - ng_l2tp.c, 1385
- ng_l2tp_config_type_fields
 - ng_l2tp.c, 1385

- NG_L2TP_CONFIG_TYPE_INFO
 - ng_l2tp.h, 1389
- ng_l2tp_constructor
 - ng_l2tp.c, 1379, 1385
- ng_l2tp_disconnect
 - ng_l2tp.c, 1380, 1386
- ng_l2tp_find_session
 - ng_l2tp.c, 1380, 1386
- NG_L2TP_HOOK_CTRL
 - ng_l2tp.h, 1390
- NG_L2TP_HOOK_LOWER
 - ng_l2tp.h, 1390
- ng_l2tp_hook_private, 310
 - conf, 310
 - nr, 310
 - ns, 310
 - stats, 310
- NG_L2TP_HOOK_SESSION_F
 - ng_l2tp.h, 1390
- NG_L2TP_HOOK_SESSION_P
 - ng_l2tp.h, 1390
- ng_l2tp_newhook
 - ng_l2tp.c, 1380, 1386
- NG_L2TP_NODE_TYPE
 - ng_l2tp.h, 1390
- ng_l2tp_private, 311
 - conf, 311
 - ctrl, 311
 - ftarget, 311
 - lower, 311
 - node, 312
 - seq, 312
 - stats, 312
- ng_l2tp_rcvdata
 - ng_l2tp.c, 1380, 1386
- ng_l2tp_rcvmsg
 - ng_l2tp.c, 1381, 1386
- ng_l2tp_rcv_ctrl
 - ng_l2tp.c, 1381
- ng_l2tp_rcv_data
 - ng_l2tp.c, 1381
- ng_l2tp_rcv_lower
 - ng_l2tp.c, 1382
- ng_l2tp_reset_session
 - ng_l2tp.c, 1382, 1386
- ng_l2tp_seq_adjust
 - ng_l2tp.c, 1382
- ng_l2tp_seq_config, 313
 - nr, 313
 - ns, 313
 - rack, 313
 - xack, 313
- ng_l2tp_seq_config_fields
 - ng_l2tp.c, 1386
- ng_l2tp_seq_config_type
 - ng_l2tp.c, 1386
- NG_L2TP_SEQ_CONFIG_TYPE_INFO
 - ng_l2tp.h, 1390
- ng_l2tp_seq_failure
 - ng_l2tp.c, 1383
- ng_l2tp_seq_init
 - ng_l2tp.c, 1383
- ng_l2tp_seq_rack_timeout
 - ng_l2tp.c, 1383
- ng_l2tp_seq_rcv_nr
 - ng_l2tp.c, 1383
- ng_l2tp_seq_rcv_ns
 - ng_l2tp.c, 1383
- ng_l2tp_seq_reset
 - ng_l2tp.c, 1384
- ng_l2tp_seq_set
 - ng_l2tp.c, 1384
- ng_l2tp_seq_xack_timeout
 - ng_l2tp.c, 1384
- ng_l2tp_sess_config, 314
 - control_dseq, 314
 - enable_dseq, 314
 - include_length, 314
 - peer_id, 314
 - session_id, 314
- ng_l2tp_sess_config_type
 - ng_l2tp.c, 1386
- ng_l2tp_sess_config_type_fields
 - ng_l2tp.c, 1387
- NG_L2TP_SESS_CONFIG_TYPE_INFO
 - ng_l2tp.h, 1390
- ng_l2tp_session_stats, 315
 - recvOctets, 315
 - recvPackets, 315
 - xmitOctets, 315
 - xmitPackets, 315
- ng_l2tp_session_stats_type
 - ng_l2tp.c, 1387
- ng_l2tp_session_stats_type_fields
 - ng_l2tp.c, 1387
- NG_L2TP_SESSION_STATS_TYPE_INFO
 - ng_l2tp.h, 1391
- ng_l2tp_shutdown
 - ng_l2tp.c, 1384, 1387
- ng_l2tp_stats, 316
 - memoryFailures, 316
 - recvBadAcks, 316
 - recvDataDrops, 316
 - recvDuplicates, 316
 - recvInvalid, 316
 - recvOctets, 317
 - recvOutOfOrder, 317
 - recvPackets, 317

- recvRunts, 317
- recvUnknownSID, 317
- recvWrongTunnel, 317
- recvZLBs, 317
- xmitDataTooBig, 317
- xmitDrops, 317
- xmitInvalid, 317
- xmitOctets, 317
- xmitPackets, 318
- xmitRetransmits, 318
- xmitTooBig, 318
- xmitZLBs, 318
- ng_l2tp_stats_type
 - ng_l2tp.c, 1387
- ng_l2tp_stats_type_fields
 - ng_l2tp.c, 1387
- NG_L2TP_STATS_TYPE_INFO
 - ng_l2tp.h, 1391
- ng_l2tp_tpestruct
 - ng_l2tp.c, 1387
- ng_l2tp_xmit_ctrl
 - ng_l2tp.c, 1385
- ng_leave_read
 - ng_base.c, 1190
- ng_leave_write
 - ng_base.c, 1190
- ng_lmi.c
 - ANNEXA, 1395
 - ANNEXD, 1395
 - DLCI_DOWN, 1395
 - DLCI_NULL, 1395
 - DLCI_UP, 1395
 - GROUP4, 1395
 - LMI_MIN_LENGTH, 1395
 - LMI_PATIENCE, 1395
 - LMI_ticker, 1398
 - LMIPOLLSIZE, 1395
 - MAX_DLCIS, 1395
 - MAXDLCI, 1396
 - NAME_ANNEXA, 1396
 - NAME_ANNEXD, 1396
 - NAME_GROUP4, 1396
 - NAME_NONE, 1396
 - NETGRAPH_INIT, 1398
 - ngauto_state_machine, 1398
 - nglmi_checkdata, 1398
 - nglmi_constructor, 1399, 1401
 - nglmi_disconnect, 1399, 1401
 - nglmi_inquire, 1399
 - nglmi_newhook, 1399, 1401
 - nglmi_rcvdata, 1400, 1401
 - nglmi_rcvmsg, 1400, 1401
 - nglmi_shutdown, 1400, 1402
 - nglmi_startup, 1400
 - nglmi_startup_auto, 1401
 - nglmi_startup_fixed, 1401
 - NOPROTO, 1396
 - sc_p, 1398
 - SCF_ANNEX_A, 1396
 - SCF_ANNEX_D, 1396
 - SCF_AUTO, 1396
 - SCF_CONNECTED, 1397
 - SCF_FIXED, 1397
 - SCF_GROUP4, 1397
 - SCF_LMITYPE, 1397
 - SCF_NOLMI, 1397
 - SETLMITYPE, 1397
 - STEPBY, 1397
 - typestruct, 1402
- ng_lmi.h
 - NGM_LMI_GET_STATUS, 1405
- ng_lmi.h
 - NG_LMI_HOOK_ANNEXA, 1403
 - NG_LMI_HOOK_ANNEXD, 1403
 - NG_LMI_HOOK_AUTO0, 1403
 - NG_LMI_HOOK_AUTO1023, 1403
 - NG_LMI_HOOK_DEBUG, 1404
 - NG_LMI_HOOK_GROUPOF4, 1404
 - NG_LMI_KEEPALIVE_RATE, 1404
 - NG_LMI_LMI_PRIORITY, 1404
 - NG_LMI_NODE_TYPE, 1404
 - NG_LMI_POLL_RATE, 1404
 - NG_LMI_SEQ_PER_FULL, 1404
 - NGM_LMI_COOKIE, 1404
 - NGM_LMI_STAT_ARYSIZE, 1404
- NG_LMI_HOOK_ANNEXA
 - ng_lmi.h, 1403
- NG_LMI_HOOK_ANNEXD
 - ng_lmi.h, 1403
- NG_LMI_HOOK_AUTO0
 - ng_lmi.h, 1403
- NG_LMI_HOOK_AUTO1023
 - ng_lmi.h, 1403
- NG_LMI_HOOK_DEBUG
 - ng_lmi.h, 1404
- NG_LMI_HOOK_GROUPOF4
 - ng_lmi.h, 1404
- NG_LMI_KEEPALIVE_RATE
 - ng_lmi.h, 1404
- NG_LMI_LMI_PRIORITY
 - ng_lmi.h, 1404
- NG_LMI_NODE_TYPE
 - ng_lmi.h, 1404
- NG_LMI_POLL_RATE
 - ng_lmi.h, 1404
- NG_LMI_SEQ_PER_FULL
 - ng_lmi.h, 1404
- ng_lostpdu

- hva_stats_ng, 93
- ng_mac_addr_type
 - ng_atmpif.c, 539
- ng_macaddr_parse
 - ng_atmpif.c, 536, 539
- ng_macaddr_unparse
 - ng_atmpif.c, 537, 539
- ng_macro_test
 - ng_base.c, 1191
- ng_make_node
 - ng_base.c, 1191
- ng_make_node_common
 - netgraph.h, 1152
 - ng_base.c, 1191
- ng_mesg, 319
 - data, 319
 - header, 320
- ng_mesg::ng_msghdr, 321
 - arglen, 321
 - cmd, 321
 - cmdstr, 322
 - flags, 322
 - spare, 322
 - spare2, 322
 - token, 322
 - typecookie, 323
 - version, 323
- ng_message.h
 - NGM_ASCII2BINARY, 1418
 - NGM_BINARY2ASCII, 1418
 - NGM_CONNECT, 1418
 - NGM_LISTHOOKS, 1418
 - NGM_LISTNAMES, 1418
 - NGM_LISTNODES, 1418
 - NGM_LISTTYPES, 1418
 - NGM_MKPEER, 1418
 - NGM_NAME, 1418
 - NGM_NODEINFO, 1418
 - NGM_RMHOOK, 1418
 - NGM_SHUTDOWN, 1418
 - NGM_TEXT_CONFIG, 1418
 - NGM_TEXT_STATUS, 1418
- ng_message.h
 - NG_CMDSTRLEN, 1409
 - NG_CMDSTRSIZ, 1409
 - NG_COPYMESSAGE, 1409
 - NG_GENERIC_BANDWIDTH_INFO, 1410
 - NG_GENERIC_CONNECT_INFO, 1410
 - NG_GENERIC_FLOW_MANAGER_INFO, 1410
 - NG_GENERIC_HOOKLIST_INFO, 1411
 - NG_GENERIC_LINKINFO_INFO, 1411
 - NG_GENERIC_LISTNODES_INFO, 1411
 - NG_GENERIC_MKPEER_INFO, 1411
 - NG_GENERIC_NAME_INFO, 1411
 - NG_GENERIC_NG_MESG_INFO, 1412
 - NG_GENERIC_NODEINFO_INFO, 1412
 - NG_GENERIC_QUEUE_INFO, 1412
 - NG_GENERIC_RMHOOK_INFO, 1412
 - NG_GENERIC_TYPEINFO_INFO, 1413
 - NG_GENERIC_TYPELIST_INFO, 1413
 - NG_HOOKLEN, 1413
 - NG_HOOKSIZ, 1413
 - ng_ID_t, 1413
 - NG_MKMESSAGE, 1413
 - NG_MKRESPONSE, 1414
 - NG_NODELEN, 1415
 - NG_NODESIZ, 1415
 - NG_PATHLEN, 1415
 - NG_PATHSIZ, 1415
 - NG_TEXTRESPONSE, 1415
 - NG_TYPELEN, 1415
 - NG_TYPESIZ, 1415
 - NG_VERSION, 1415
 - NGF_ORIG, 1415
 - NGF_RESP, 1416
 - NGIOCGINFO, 1416
 - NGIOCSETNAME, 1416
 - NGM_DROP_LINK, 1416
 - NGM_FLOW_COOKIE, 1416
 - NGM_FLUSH_QUEUE, 1416
 - NGM_GENERIC_COOKIE, 1416
 - NGM_GET_BANDWIDTH, 1416
 - NGM_GET_XMIT_Q_LIMITS, 1417
 - NGM_HASREPLY, 1417
 - NGM_HIGH_WATER_PASSED, 1417
 - NGM_LINK_IS_DOWN, 1417
 - NGM_LINK_IS_UP, 1417
 - NGM_LOW_WATER_PASSED, 1417
 - NGM_MICROMANAGE, 1417
 - NGM_RAISE_LINK, 1417
 - NGM_READONLY, 1417
 - NGM_SET_FLOW_MANAGER, 1418
 - NGM_SET_XMIT_Q_LIMITS, 1418
 - NGM_SYNC_QUEUE_STATE, 1418
- NG_MKMESSAGE
 - ng_message.h, 1413
- ng_mkpeer
 - ng_base.c, 1191
- NG_MKRESPONSE
 - ng_message.h, 1414
- ng_mod_event
 - netgraph.h, 1153
 - ng_base.c, 1192
- ng_mppc.c
 - ERRROUT, 1421
 - KEYLEN, 1421
 - M_NETGRAPH_MPPC, 1421

- MPPC_CCOUNT_MASK, 1421
- MPPC_COMP_OK, 1421
- MPPC_DECOMP_BUFSIZE, 1421
- MPPC_DECOMP_OK, 1421
- MPPC_DECOMP_SAFETY, 1421
- MPPC_FLAG_COMPRESSED, 1421
- MPPC_FLAG_ENCRYPTED, 1422
- MPPC_FLAG_FLUSHED, 1422
- MPPC_FLAG_RESTART, 1422
- MPPC_HDRLLEN, 1422
- MPPE_MAX_REKEY, 1422
- MPPE_UPDATE_FLAG, 1422
- MPPE_UPDATE_MASK, 1422
- NETGRAPH_INIT, 1423
- ng_mppc_compress, 1423
- ng_mppc_constructor, 1423, 1425
- ng_mppc_decompress, 1423
- ng_mppc_disconnect, 1423, 1425
- ng_mppc_getkey, 1424
- ng_mppc_newhook, 1424, 1425
- ng_mppc_rcvdata, 1424, 1425
- ng_mppc_rcvmsg, 1424, 1426
- ng_mppc_reset_req, 1425
- ng_mppc_shutdown, 1425, 1426
- ng_mppc_tpestruct, 1426
- ng_mppc_updatekey, 1425
- ng_mppe_weakenkey, 1426
- priv_p, 1423
- ng_mppc.h
 - NGM_MPPC_CONFIG_COMP, 1429
 - NGM_MPPC_CONFIG_DECOMP, 1429
 - NGM_MPPC_RESETRREQ, 1429
- ng_mppc.c
 - MPPC_BIT, 1427
 - MPPC_MAX_BLOWUP, 1427
 - MPPC_VALID_BITS, 1427
 - MPPE_128, 1428
 - MPPE_40, 1428
 - MPPE_56, 1428
 - MPPE_BITS, 1428
 - MPPE_KEY_LEN, 1428
 - MPPE_STATELESS, 1428
 - NG_MPPC_HOOK_COMP, 1428
 - NG_MPPC_HOOK_DECOMP, 1428
 - NG_MPPC_NODE_TYPE, 1428
 - NGM_MPPC_COOKIE, 1429
- ng_mppc_compress
 - ng_mppc.c, 1423
- ng_mppc_config, 324
 - bits, 324
 - enable, 324
 - startkey, 324
- ng_mppc_constructor
 - ng_mppc.c, 1423, 1425
- ng_mppc_decompress
 - ng_mppc.c, 1423
- ng_mppc_dir, 325
 - cc, 325
 - cfg, 325
 - flushed, 325
 - hook, 325
- ng_mppc_disconnect
 - ng_mppc.c, 1423, 1425
- ng_mppc_getkey
 - ng_mppc.c, 1424
- NG_MPPC_HOOK_COMP
 - ng_mppc.h, 1428
- NG_MPPC_HOOK_DECOMP
 - ng_mppc.h, 1428
- ng_mppc_newhook
 - ng_mppc.c, 1424, 1425
- NG_MPPC_NODE_TYPE
 - ng_mppc.h, 1428
- ng_mppc_private, 327
 - ctrlnode, 327
 - recv, 327
 - xmit, 327
- ng_mppc_rcvdata
 - ng_mppc.c, 1424, 1425
- ng_mppc_rcvmsg
 - ng_mppc.c, 1424, 1426
- ng_mppc_reset_req
 - ng_mppc.c, 1425
- ng_mppc_shutdown
 - ng_mppc.c, 1425, 1426
- ng_mppc_tpestruct
 - ng_mppc.c, 1426
- ng_mppc_updatekey
 - ng_mppc.c, 1425
- ng_mppe_weakenkey
 - ng_mppc.c, 1426
- ng_msg_data_type
 - ng_parse.c, 1466
- ng_name2noderef
 - ng_base.c, 1192
- ng_name_node
 - netgraph.h, 1153
 - ng_base.c, 1192
- ng_nat.c
 - MODULE_DEPEND, 1432
 - NETGRAPH_INIT, 1432
 - ng_nat_cmdlist, 1433
 - ng_nat_constructor, 1432, 1434
 - ng_nat_disconnect, 1432, 1434
 - ng_nat_newhook, 1433, 1434
 - ng_nat_rcvdata, 1433, 1434
 - ng_nat_rcvmsg, 1433, 1434
 - ng_nat_shutdown, 1433, 1434

- NGNAT_ADDR_DEFINED, 1432
- NGNAT_READY, 1432
- priv_p, 1432
- typestruct, 1434
- ng_nat.h
 - NGM_NAT_SET_IPADDR, 1435
- ng_nat.h
 - NG_NAT_HOOK_IN, 1435
 - NG_NAT_HOOK_OUT, 1435
 - NG_NAT_NODE_TYPE, 1435
 - NGM_NAT_COOKIE, 1435
- ng_nat_cmdlist
 - ng_nat.c, 1433
- ng_nat_constructor
 - ng_nat.c, 1432, 1434
- ng_nat_disconnect
 - ng_nat.c, 1432, 1434
- NG_NAT_HOOK_IN
 - ng_nat.h, 1435
- NG_NAT_HOOK_OUT
 - ng_nat.h, 1435
- ng_nat_newhook
 - ng_nat.c, 1433, 1434
- NG_NAT_NODE_TYPE
 - ng_nat.h, 1435
- ng_nat_priv, 328
 - flags, 328
 - in, 328
 - lib, 328
 - node, 329
 - out, 329
- ng_nat_rcvdata
 - ng_nat.c, 1433, 1434
- ng_nat_rcvmsg
 - ng_nat.c, 1433, 1434
- ng_nat_shutdown
 - ng_nat.c, 1433, 1434
- ng_netflow.c
 - M_CHECK, 1102
 - NETGRAPH_INIT, 1103
 - ng_netflow_close, 1103, 1105
 - ng_netflow_cmds, 1105
 - ng_netflow_constructor, 1103, 1105
 - ng_netflow_disconnect, 1103, 1105
 - ng_netflow_ifinfo_type, 1105
 - ng_netflow_ifinfo_type_fields, 1105
 - ng_netflow_info_type, 1105
 - ng_netflow_info_type_fields, 1105
 - ng_netflow_newhook, 1103, 1105
 - ng_netflow_rcvdata, 1104, 1106
 - ng_netflow_rcvmsg, 1104, 1106
 - ng_netflow_rmnode, 1104, 1106
 - ng_netflow_setdlt_type, 1106
 - ng_netflow_setdlt_type_fields, 1106
 - ng_netflow_setifindex_type, 1106
 - ng_netflow_setifindex_type_fields, 1106
 - ng_netflow_settimeouts_type, 1106
 - ng_netflow_settimeouts_type_fields, 1107
 - ng_netflow_typestruct, 1107
 - rds_id, 1107
- ng_netflow.h
 - NGM_NETFLOW_IFINFO, 1113
 - NGM_NETFLOW_INFO, 1113
 - NGM_NETFLOW_SETDLT, 1113
 - NGM_NETFLOW_SETIFINDEX, 1113
 - NGM_NETFLOW_SETTIMEOUTS, 1113
 - NGM_NETFLOW_SHOW, 1113
- ng_netflow.h
 - CACHEHIGHWAT, 1109
 - CACHELOWAT, 1109
 - CACHESIZE, 1109
 - ERRROUT, 1109
 - fle_i_ifx, 1110
 - iface_p, 1113
 - ifinfo_p, 1113
 - MAXDLTNAMELEN, 1110
 - ng_netflow_cache_flush, 1114
 - ng_netflow_cache_init, 1114
 - ng_netflow_copyinfo, 1114
 - ng_netflow_expire, 1115
 - ng_netflow_flow_add, 1114
 - ng_netflow_flow_show, 1115
 - NG_NETFLOW_HOOK_DATA, 1110
 - NG_NETFLOW_HOOK_EXPORT, 1110
 - NG_NETFLOW_HOOK_OUT, 1110
 - NG_NETFLOW_IFINFO_TYPE, 1110
 - NG_NETFLOW_INFO_TYPE, 1110
 - NG_NETFLOW_MAXIFACES, 1111
 - NG_NETFLOW_NODE_TYPE, 1111
 - NG_NETFLOW_SETDLT_TYPE, 1111
 - NG_NETFLOW_SETIFINDEX_TYPE, 1111
 - NG_NETFLOW_SETTIMEOUTS_TYPE, 1111
 - NGM_NETFLOW_COOKIE, 1112
 - NGRESP_SIZE, 1112
 - NREC_AT_ONCE, 1112
 - priv_p, 1113
 - r_dport, 1112
 - r_i_ifx, 1112
 - r_ip_p, 1112
 - r_misc, 1112
 - r_ports, 1112
 - r_sport, 1113
 - r_tos, 1113
 - SORCVBUF_SIZE, 1113
- ng_netflow_cache_flush
 - netflow.c, 1095
 - ng_netflow.h, 1114

- ng_netflow_cache_init
 - netflow.c, 1095
 - ng_netflow.h, 1114
- ng_netflow_close
 - ng_netflow.c, 1103, 1105
- ng_netflow_cmds
 - ng_netflow.c, 1105
- ng_netflow_constructor
 - ng_netflow.c, 1103, 1105
- ng_netflow_copyinfo
 - netflow.c, 1095
 - ng_netflow.h, 1114
- ng_netflow_disconnect
 - ng_netflow.c, 1103, 1105
- ng_netflow_expire
 - netflow.c, 1096
 - ng_netflow.h, 1115
- ng_netflow_flow_add
 - netflow.c, 1096
 - ng_netflow.h, 1114
- ng_netflow_flow_show
 - netflow.c, 1096
 - ng_netflow.h, 1115
- NG_NETFLOW_HOOK_DATA
 - ng_netflow.h, 1110
- NG_NETFLOW_HOOK_EXPORT
 - ng_netflow.h, 1110
- NG_NETFLOW_HOOK_OUT
 - ng_netflow.h, 1110
- ng_netflow_iface, 330
 - hook, 330
 - info, 330
 - out, 330
- ng_netflow_ifinfo, 332
 - ifinfo_dlt, 332
 - ifinfo_index, 332
 - ifinfo_packets, 332
- NG_NETFLOW_IFINFO_TYPE
 - ng_netflow.h, 1110
- ng_netflow_ifinfo_type
 - ng_netflow.c, 1105
- ng_netflow_ifinfo_type_fields
 - ng_netflow.c, 1105
- ng_netflow_info, 333
 - ninfo_act_exp, 333
 - ninfo_act_t, 333
 - ninfo_alloc_failed, 333
 - ninfo_bytes, 333
 - ninfo_export_failed, 333
 - ninfo_inact_exp, 333
 - ninfo_inact_t, 333
 - ninfo_packets, 334
 - ninfo_used, 334
- NG_NETFLOW_INFO_TYPE
 - ng_netflow.h, 1110
- ng_netflow_info_type
 - ng_netflow.c, 1105
- ng_netflow_info_type_fields
 - ng_netflow.c, 1105
- NG_NETFLOW_MAXIFACES
 - ng_netflow.h, 1111
- ng_netflow_newhook
 - ng_netflow.c, 1103, 1105
- NG_NETFLOW_NODE_TYPE
 - ng_netflow.h, 1111
- ng_netflow_rcvdata
 - ng_netflow.c, 1104, 1106
- ng_netflow_rcvmsg
 - ng_netflow.c, 1104, 1106
- ng_netflow_rmnode
 - ng_netflow.c, 1104, 1106
- ng_netflow_setdlt, 335
 - dlt, 335
 - iface, 335
- NG_NETFLOW_SETDLT_TYPE
 - ng_netflow.h, 1111
- ng_netflow_setdlt_type
 - ng_netflow.c, 1106
- ng_netflow_setdlt_type_fields
 - ng_netflow.c, 1106
- ng_netflow_setifindex, 336
 - iface, 336
 - index, 336
- NG_NETFLOW_SETIFINDEX_TYPE
 - ng_netflow.h, 1111
- ng_netflow_setifindex_type
 - ng_netflow.c, 1106
- ng_netflow_setifindex_type_fields
 - ng_netflow.c, 1106
- ng_netflow_settimeouts, 337
 - active_timeout, 337
 - inactive_timeout, 337
- NG_NETFLOW_SETTIMEOUTS_TYPE
 - ng_netflow.h, 1111
- ng_netflow_settimeouts_type
 - ng_netflow.c, 1106
- ng_netflow_settimeouts_type_fields
 - ng_netflow.c, 1107
- ng_netflow_typestruct
 - ng_netflow.c, 1107
- ng_newhook_t
 - netgraph.h, 1150
- ng_newtype
 - netgraph.h, 1153
 - ng_base.c, 1193
- ng_node, 338
 - LIST_ENTRY, 339
 - LIST_HEAD, 339

- nd_flags, 339
- nd_ID, 339
- nd_input_queue, 339
- nd_name, 339
- nd_numhooks, 339
- nd_private, 339
- nd_refs, 339
- nd_type, 340
- TAILQ_ENTRY, 339
- ng_node2ID
 - netgraph.h, 1154
 - ng_base.c, 1193
- NG_NODE_FORCE_WRITER
 - netgraph.h, 1134
- NG_NODE_FOREACH_HOOK
 - netgraph.h, 1135
- NG_NODE_HAS_NAME
 - netgraph.h, 1135
- NG_NODE_ID
 - netgraph.h, 1135
- NG_NODE_IS_VALID
 - netgraph.h, 1135
- NG_NODE_NAME
 - netgraph.h, 1135
- NG_NODE_NOT_VALID
 - netgraph.h, 1136
- NG_NODE_NUMHOOKS
 - netgraph.h, 1136
- NG_NODE_PRIVATE
 - netgraph.h, 1136
- NG_NODE_REALLY_DIE
 - netgraph.h, 1138
- NG_NODE_REF
 - netgraph.h, 1138
- NG_NODE_REVIVE
 - netgraph.h, 1138
- NG_NODE_SET_PRIVATE
 - netgraph.h, 1138
- NG_NODE_UNREF
 - netgraph.h, 1139
- ng_nodeinfoarray_type_info
 - ng_base.c, 1203
- NG_NODELEN
 - ng_message.h, 1415
- NG_NODESIZ
 - ng_message.h, 1415
- NG_NOFLAGS
 - netgraph.h, 1139
- ng_one2many.c
 - NETGRAPH_INIT, 1438
 - ng_one2many_cmdlist, 1440
 - ng_one2many_config_type, 1440
 - ng_one2many_config_type_fields, 1440
 - ng_one2many_constructor, 1438, 1440
 - ng_one2many_disconnect, 1438, 1441
 - ng_one2many_enableLinks_array_type, 1441
 - ng_one2many_enableLinks_array_type_info, 1441
 - ng_one2many_link_stats_type, 1441
 - ng_one2many_link_stats_type_fields, 1441
 - ng_one2many_newhook, 1439, 1441
 - ng_one2many_notify, 1439
 - ng_one2many_rcvdata, 1439, 1441
 - ng_one2many_rcvmsg, 1439, 1442
 - ng_one2many_shutdown, 1440, 1442
 - ng_one2many_tpestruct, 1442
 - ng_one2many_update_many, 1440
 - priv_p, 1438
- ng_one2many.h
 - NGM_ONE2MANY_CLR_STATS, 1445
 - NGM_ONE2MANY_GET_CONFIG, 1445
 - NGM_ONE2MANY_GET_STATS, 1445
 - NGM_ONE2MANY_GETCLR_STATS, 1445
 - NGM_ONE2MANY_SET_CONFIG, 1445
- ng_one2many.h
 - NG_ONE2MANY_CONFIG_TYPE_INFO, 1443
 - NG_ONE2MANY_FAIL_MANUAL, 1443
 - NG_ONE2MANY_FAIL_NOTIFY, 1444
 - NG_ONE2MANY_HOOK_MANY_FMT, 1444
 - NG_ONE2MANY_HOOK_MANY_PREFIX, 1444
 - NG_ONE2MANY_HOOK_ONE, 1444
 - NG_ONE2MANY_LINK_STATS_TYPE_INFO, 1444
 - NG_ONE2MANY_MAX_LINKS, 1444
 - NG_ONE2MANY_NODE_TYPE, 1444
 - NG_ONE2MANY_ONE_LINKNUM, 1444
 - NG_ONE2MANY_XMIT_ALL, 1445
 - NG_ONE2MANY_XMIT_ROUNDROBIN, 1445
 - NGM_ONE2MANY_COOKIE, 1445
- ng_one2many_cmdlist
 - ng_one2many.c, 1440
- ng_one2many_config, 341
 - enabledLinks, 341
 - failAlg, 341
 - xmitAlg, 341
- ng_one2many_config_type
 - ng_one2many.c, 1440
- ng_one2many_config_type_fields
 - ng_one2many.c, 1440
- NG_ONE2MANY_CONFIG_TYPE_INFO
 - ng_one2many.h, 1443
- ng_one2many_constructor
 - ng_one2many.c, 1438, 1440
- ng_one2many_disconnect

- ng_one2many.c, [1438](#), [1441](#)
- ng_one2many_enableLinks_array_type
 - ng_one2many.c, [1441](#)
- ng_one2many_enableLinks_array_type_info
 - ng_one2many.c, [1441](#)
- NG_ONE2MANY_FAIL_MANUAL
 - ng_one2many.h, [1443](#)
- NG_ONE2MANY_FAIL_NOTIFY
 - ng_one2many.h, [1444](#)
- NG_ONE2MANY_HOOK_MANY_FMT
 - ng_one2many.h, [1444](#)
- NG_ONE2MANY_HOOK_MANY_PREFIX
 - ng_one2many.h, [1444](#)
- NG_ONE2MANY_HOOK_ONE
 - ng_one2many.h, [1444](#)
- ng_one2many_link, [342](#)
 - hook, [342](#)
 - stats, [342](#)
- ng_one2many_link_stats, [343](#)
 - memoryFailures, [343](#)
 - recvOctets, [343](#)
 - recvPackets, [343](#)
 - xmitOctets, [343](#)
 - xmitPackets, [343](#)
- ng_one2many_link_stats_type
 - ng_one2many.c, [1441](#)
- ng_one2many_link_stats_type_fields
 - ng_one2many.c, [1441](#)
- NG_ONE2MANY_LINK_STATS_TYPE_INFO
 - ng_one2many.h, [1444](#)
- NG_ONE2MANY_MAX_LINKS
 - ng_one2many.h, [1444](#)
- ng_one2many_newhook
 - ng_one2many.c, [1439](#), [1441](#)
- NG_ONE2MANY_NODE_TYPE
 - ng_one2many.h, [1444](#)
- ng_one2many_notify
 - ng_one2many.c, [1439](#)
- NG_ONE2MANY_ONE_LINKNUM
 - ng_one2many.h, [1444](#)
- ng_one2many_private, [344](#)
 - activeMany, [344](#)
 - conf, [344](#)
 - many, [344](#)
 - nextMany, [345](#)
 - node, [345](#)
 - numActiveMany, [345](#)
 - one, [345](#)
- ng_one2many_rcvdata
 - ng_one2many.c, [1439](#), [1441](#)
- ng_one2many_rcvmsg
 - ng_one2many.c, [1439](#), [1442](#)
- ng_one2many_shutdown
 - ng_one2many.c, [1440](#), [1442](#)
- ng_one2many_tpestruct
 - ng_one2many.c, [1442](#)
- ng_one2many_update_many
 - ng_one2many.c, [1440](#)
- NG_ONE2MANY_XMIT_ALL
 - ng_one2many.h, [1445](#)
- NG_ONE2MANY_XMIT_ROUNDROBIN
 - ng_one2many.h, [1445](#)
- ng_package_data
 - netgraph.h, [1154](#)
 - ng_base.c, [1193](#)
- ng_package_msg
 - netgraph.h, [1154](#)
 - ng_base.c, [1193](#)
- ng_package_msg_self
 - netgraph.h, [1154](#)
 - ng_base.c, [1194](#)
- ng_parse
 - ng_parse.c, [1461](#)
 - ng_parse.h, [1479](#)
- ng_parse.c
 - CT_ARRAY, [1452](#)
 - CT_FIXEDARRAY, [1452](#)
 - CT_STRUCT, [1452](#)
- ng_parse.c
 - ALIGNMENT, [1451](#)
 - comptype, [1452](#)
 - INT16_ALIGNMENT, [1451](#)
 - INT32_ALIGNMENT, [1451](#)
 - INT64_ALIGNMENT, [1451](#)
 - INT8_ALIGNMENT, [1451](#)
 - INT_HEX, [1451](#)
 - INT_SIGNED, [1451](#)
 - INT_UNSIGNED, [1451](#)
 - INVOKE, [1452](#)
 - M_NETGRAPH_PARSE, [1452](#)
 - METHOD, [1452](#)
 - ng_array_getAlign, [1452](#)
 - ng_array_getDefault, [1452](#)
 - ng_array_parse, [1453](#)
 - ng_array_unparse, [1453](#)
 - ng_bytearray_getDefault, [1454](#)
 - ng_bytearray_parse, [1454](#)
 - ng_bytearray_unparse, [1454](#)
 - ng_enaddr_parse, [1454](#)
 - ng_enaddr_unparse, [1455](#)
 - ng_encode_string, [1455](#)
 - ng_fixedarray_getAlign, [1455](#)
 - ng_fixedarray_getDefault, [1455](#)
 - ng_fixedarray_parse, [1455](#)
 - ng_fixedarray_unparse, [1456](#)
 - ng_fixedstring_getDefault, [1456](#)
 - ng_fixedstring_parse, [1456](#)
 - ng_fixedstring_unparse, [1456](#)

- ng_get_composite_elem_default, 1457
- ng_get_composite_etype, 1457
- ng_get_composite_len, 1457
- ng_get_getAlign_method, 1457
- ng_get_getDefault_method, 1458
- ng_get_parse_method, 1458
- ng_get_string_token, 1458
- ng_get_unparse_method, 1458
- ng_int16_getAlign, 1458
- ng_int16_getDefault, 1458
- ng_int16_parse, 1458
- ng_int16_unparse, 1458
- ng_int32_getAlign, 1459
- ng_int32_getDefault, 1459
- ng_int32_parse, 1459
- ng_int32_unparse, 1459
- ng_int64_getAlign, 1459
- ng_int64_getDefault, 1459
- ng_int64_parse, 1459
- ng_int64_unparse, 1460
- ng_int8_getAlign, 1460
- ng_int8_getDefault, 1460
- ng_int8_parse, 1460
- ng_int8_unparse, 1460
- ng_ipaddr_getDefault, 1460
- ng_ipaddr_parse, 1461
- ng_ipaddr_unparse, 1461
- ng_msg_data_type, 1466
- ng_parse, 1461
- ng_parse_append, 1461
- ng_parse_array_type, 1466
- ng_parse_bytearray_subtype, 1467
- ng_parse_bytearray_subtype_getLength, 1461
- ng_parse_bytearray_subtype_info, 1467
- ng_parse_bytearray_type, 1467
- ng_parse_cmdbuf_info, 1467
- ng_parse_cmdbuf_type, 1468
- ng_parse_composite, 1461
- ng_parse_enaddr_type, 1468
- ng_parse_fixedarray_type, 1468
- ng_parse_fixedstring_type, 1468
- ng_parse_get_elem_pad, 1462
- ng_parse_get_token, 1462
- ng_parse_getDefault, 1463
- ng_parse_hint16_type, 1469
- ng_parse_hint32_type, 1469
- ng_parse_hint64_type, 1469
- ng_parse_hint8_type, 1469
- ng_parse_hookbuf_info, 1470
- ng_parse_hookbuf_type, 1470
- ng_parse_int16_type, 1470
- ng_parse_int32_type, 1470
- ng_parse_int64_type, 1471
- ng_parse_int8_type, 1471
- ng_parse_ipaddr_type, 1471
- ng_parse_ng_msg_getLength, 1463
- ng_parse_ng_msg_type, 1471
- ng_parse_ng_msg_type_fields, 1472
- ng_parse_nodebuf_info, 1472
- ng_parse_nodebuf_type, 1472
- ng_parse_pathbuf_info, 1472
- ng_parse_pathbuf_type, 1472
- ng_parse_sizedstring_type, 1472
- ng_parse_skip_value, 1463
- ng_parse_string_type, 1473
- ng_parse_struct_type, 1473
- ng_parse_typebuf_info, 1473
- ng_parse_typebuf_type, 1473
- ng_parse_uint16_type, 1474
- ng_parse_uint32_type, 1474
- ng_parse_uint64_type, 1474
- ng_parse_uint8_type, 1474
- ng_sizedstring_getDefault, 1463
- ng_sizedstring_parse, 1463
- ng_sizedstring_unparse, 1463
- ng_string_getDefault, 1464
- ng_string_parse, 1464
- ng_string_unparse, 1464
- ng_struct_getAlign, 1464
- ng_struct_getDefault, 1465
- ng_struct_parse, 1465
- ng_struct_unparse, 1465
- ng_unparse, 1466
- ng_unparse_composite, 1466
- ng_parse.h
 - T_EOF, 1479
 - T_EQUALS, 1479
 - T_ERROR, 1479
 - T_LBRACE, 1479
 - T_LBRACKET, 1479
 - T_RBRACE, 1479
 - T_RBRACKET, 1479
 - T_STRING, 1479
 - T_WORD, 1479
- ng_parse.h
 - ng_encode_string, 1479
 - ng_get_string_token, 1479
 - ng_getAlign_t, 1478
 - ng_getDefault_t, 1478
 - ng_parse, 1479
 - ng_parse_array_getDefault_t, 1478
 - ng_parse_array_getLength_t, 1478
 - ng_parse_array_type, 1480
 - ng_parse_bytearray_type, 1480
 - ng_parse_cmdbuf_type, 1480
 - ng_parse_enaddr_type, 1480
 - ng_parse_fixedarray_type, 1480
 - ng_parse_fixedstring_type, 1480

- ng_parse_get_token, 1479
- ng_parse_getDefault, 1480
- ng_parse_hint16_type, 1481
- ng_parse_hint32_type, 1481
- ng_parse_hint64_type, 1481
- ng_parse_hint8_type, 1481
- ng_parse_hookbuf_type, 1481
- ng_parse_int16_type, 1481
- ng_parse_int32_type, 1481
- ng_parse_int64_type, 1481
- ng_parse_int8_type, 1481
- ng_parse_ipaddr_type, 1481
- ng_parse_ng_mesg_type, 1481
- ng_parse_nodebuf_type, 1482
- ng_parse_pathbuf_type, 1482
- ng_parse_sizedstring_type, 1482
- ng_parse_string_type, 1482
- ng_parse_struct_type, 1482
- ng_parse_t, 1478
- ng_parse_token, 1479
- ng_parse_typebuf_type, 1482
- ng_parse_uint16_type, 1482
- ng_parse_uint32_type, 1482
- ng_parse_uint64_type, 1482
- ng_parse_uint8_type, 1482
- ng_unparse, 1480
- ng_unparse_t, 1478
- ng_parse_append
 - ng_parse.c, 1461
- ng_parse_array_getDefault_t
 - ng_parse.h, 1478
- ng_parse_array_getLength_t
 - ng_parse.h, 1478
- ng_parse_array_info, 346
 - elementType, 346
 - getDefault, 346
 - getLength, 346
- ng_parse_array_type
 - ng_parse.c, 1466
 - ng_parse.h, 1480
- ng_parse_bytearray_subtype
 - ng_parse.c, 1467
- ng_parse_bytearray_subtype_getLength
 - ng_parse.c, 1461
- ng_parse_bytearray_subtype_info
 - ng_parse.c, 1467
- ng_parse_bytearray_type
 - ng_parse.c, 1467
 - ng_parse.h, 1480
- ng_parse_cmdbuf_info
 - ng_parse.c, 1467
- ng_parse_cmdbuf_type
 - ng_parse.c, 1468
 - ng_parse.h, 1480
- ng_parse_composite
 - ng_parse.c, 1461
- ng_parse_enaddr_type
 - ng_parse.c, 1468
 - ng_parse.h, 1480
- ng_parse_fixedarray_info, 347
 - elementType, 347
 - getDefault, 347
 - length, 347
- ng_parse_fixedarray_type
 - ng_parse.c, 1468
 - ng_parse.h, 1480
- ng_parse_fixedstring_info, 348
 - bufSize, 348
- ng_parse_fixedstring_type
 - ng_parse.c, 1468
 - ng_parse.h, 1480
- ng_parse_generic_sockaddr_type_fields
 - ng_ksocket.c, 1371
- ng_parse_generic_sockdata_getLength
 - ng_ksocket.c, 1367
- ng_parse_get_elem_pad
 - ng_parse.c, 1462
- ng_parse_get_token
 - ng_parse.c, 1462
 - ng_parse.h, 1479
- ng_parse_getDefault
 - ng_parse.c, 1463
 - ng_parse.h, 1480
- ng_parse_hint16_type
 - ng_parse.c, 1469
 - ng_parse.h, 1481
- ng_parse_hint32_type
 - ng_parse.c, 1469
 - ng_parse.h, 1481
- ng_parse_hint64_type
 - ng_parse.c, 1469
 - ng_parse.h, 1481
- ng_parse_hint8_type
 - ng_parse.c, 1469
 - ng_parse.h, 1481
- ng_parse_hookbuf_info
 - ng_parse.c, 1470
- ng_parse_hookbuf_type
 - ng_parse.c, 1470
 - ng_parse.h, 1481
- ng_parse_int16_type
 - ng_parse.c, 1470
 - ng_parse.h, 1481
- ng_parse_int32_type
 - ng_parse.c, 1470
 - ng_parse.h, 1481
- ng_parse_int64_type
 - ng_parse.c, 1471

- ng_parse.h, 1481
- ng_parse_int8_type
 - ng_parse.c, 1471
 - ng_parse.h, 1481
- ng_parse_ipaddr_type
 - ng_parse.c, 1471
 - ng_parse.h, 1481
- ng_parse_ng_mesg_getLength
 - ng_parse.c, 1463
- ng_parse_ng_mesg_type
 - ng_parse.c, 1471
 - ng_parse.h, 1481
- ng_parse_ng_mesg_type_fields
 - ng_parse.c, 1472
- ng_parse_nodebuf_info
 - ng_parse.c, 1472
- ng_parse_nodebuf_type
 - ng_parse.c, 1472
 - ng_parse.h, 1482
- ng_parse_pathbuf_info
 - ng_parse.c, 1472
- ng_parse_pathbuf_type
 - ng_parse.c, 1472
 - ng_parse.h, 1482
- ng_parse_sizedstring_type
 - ng_parse.c, 1472
 - ng_parse.h, 1482
- ng_parse_skip_value
 - ng_parse.c, 1463
- ng_parse_sockoptval_getLength
 - ng_socket.c, 1367
- ng_parse_string_type
 - ng_parse.c, 1473
 - ng_parse.h, 1482
- ng_parse_struct_field, 349
 - alignment, 349
 - name, 349
 - type, 349
- ng_parse_struct_type
 - ng_parse.c, 1473
 - ng_parse.h, 1482
- ng_parse_t
 - ng_parse.h, 1478
- ng_parse_token
 - ng_parse.h, 1479
- ng_parse_type, 350
 - getAlign, 350
 - getDefault, 350
 - info, 350
 - parse, 350
 - private, 351
 - supertype, 351
 - unparse, 351
- ng_parse_typebuf_info
 - ng_parse.c, 1473
- ng_parse_typebuf_type
 - ng_parse.c, 1473
 - ng_parse.h, 1482
- ng_parse_uint16_type
 - ng_parse.c, 1474
 - ng_parse.h, 1482
- ng_parse_uint32_type
 - ng_parse.c, 1474
 - ng_parse.h, 1482
- ng_parse_uint64_type
 - ng_parse.c, 1474
 - ng_parse.h, 1482
- ng_parse_uint8_type
 - ng_parse.c, 1474
 - ng_parse.h, 1482
- ng_path2noderef
 - ng_base.c, 1194
- ng_path_parse
 - ng_base.c, 1194
- NG_PATHLEN
 - ng_message.h, 1415
- NG_PATHSIZ
 - ng_message.h, 1415
- NG_PEER_HOOK_NAME
 - netgraph.h, 1139
- NG_PEER_NODE
 - netgraph.h, 1140
- NG_PEER_NODE_NAME
 - netgraph.h, 1140
- ng_ppp.c
 - ERROUT, 1487
 - fn, 1506
 - HOOK_INDEX_ATALK, 1487
 - HOOK_INDEX_BYPASS, 1487
 - HOOK_INDEX_COMPRESS, 1487
 - HOOK_INDEX_DECOMPRESS, 1487
 - HOOK_INDEX_DECRYPT, 1487
 - HOOK_INDEX_ENCRYPT, 1487
 - HOOK_INDEX_INET, 1487
 - HOOK_INDEX_IPV6, 1487
 - HOOK_INDEX_IPX, 1487
 - HOOK_INDEX_MAX, 1488
 - HOOK_INDEX_VJC_COMP, 1488
 - HOOK_INDEX_VJC_IP, 1488
 - HOOK_INDEX_VJC_UNCOMP, 1488
 - HOOK_INDEX_VJC_VJIP, 1488
 - M_NETGRAPH_PPP, 1488
 - MP_FRAGTIMER_INTERVAL, 1488
 - MP_INITIAL_SEQ, 1488
 - MP_LONG_EXTEND, 1488
 - MP_LONG_FIRST_FLAG, 1489
 - MP_LONG_LAST_FLAG, 1489
 - MP_LONG_SEQ_DIFF, 1489

- MP_LONG_SEQ_HIBIT, 1489
- MP_LONG_SEQ_MASK, 1489
- MP_MAX_QUEUE_LEN, 1489
- MP_MIN_FRAG_LEN, 1489
- MP_MIN_LINK_MRU, 1489
- MP_MIN_MRRU, 1489
- MP_NEXT_RECV_SEQ, 1490
- MP_NOSEQ, 1490
- MP_RECV_SEQ_DIFF, 1490
- MP_SHORT_EXTEND, 1490
- MP_SHORT_FIRST_FLAG, 1490
- MP_SHORT_LAST_FLAG, 1490
- MP_SHORT_SEQ_DIFF, 1491
- MP_SHORT_SEQ_HIBIT, 1491
- MP_SHORT_SEQ_MASK, 1491
- name, 1506
- NETGRAPH_INIT, 1493
- ng_ppp_acf, 1506
- ng_ppp_addproto, 1493
- ng_ppp_array_info, 1506
- ng_ppp_bund_type, 1506
- ng_ppp_bund_type_fields, 1506
- ng_ppp_bypass, 1493
- ng_ppp_check_packet, 1493
- ng_ppp_cmds, 1506
- ng_ppp_comp_recv, 1493
- ng_ppp_comp_xmit, 1494
- ng_ppp_conf_type, 1506
- ng_ppp_conf_type_fields, 1507
- ng_ppp_config_valid, 1494
- ng_ppp_constructor, 1494, 1507
- ng_ppp_crypt_recv, 1494
- ng_ppp_crypt_xmit, 1495
- ng_ppp_cutproto, 1495
- ng_ppp_disconnect, 1495, 1507
- ng_ppp_frag_checkstale, 1496
- ng_ppp_frag_process, 1496
- ng_ppp_frag_reset, 1496
- ng_ppp_frag_timeout, 1497
- ng_ppp_frag_trim, 1497
- ng_ppp_get_packet, 1497
- ng_ppp_hcomp_recv, 1497
- ng_ppp_hcomp_xmit, 1498
- ng_ppp_hook_names, 1507
- ng_ppp_intcmp, 1498
- ng_ppp_link_array_type, 1507
- ng_ppp_link_type, 1507
- ng_ppp_link_type_fields, 1507
- ng_ppp_link_xmit, 1498
- ng_ppp_max_staleness, 1508
- ng_ppp_mp_recv, 1498
- ng_ppp_mp_state_type, 1508
- ng_ppp_mp_state_type_fields, 1508
- ng_ppp_mp_strategy, 1499
- ng_ppp_mp_xmit, 1499
- ng_ppp_newhook, 1500, 1508
- ng_ppp_prepend, 1500
- ng_ppp_proto_recv, 1500
- ng_ppp_rcvdata, 1500, 1508
- ng_ppp_rcvdata_atalk, 1501, 1508
- ng_ppp_rcvdata_bypass, 1501, 1508
- ng_ppp_rcvdata_compress, 1501, 1508
- ng_ppp_rcvdata_decompress, 1501, 1508
- ng_ppp_rcvdata_decrypt, 1502, 1508
- ng_ppp_rcvdata_encrypt, 1502, 1509
- ng_ppp_rcvdata_inet, 1502, 1509
- ng_ppp_rcvdata_ipv6, 1503, 1509
- ng_ppp_rcvdata_ipx, 1503, 1509
- ng_ppp_rcvdata_vjc_comp, 1503, 1509
- ng_ppp_rcvdata_vjc_ip, 1503, 1509
- ng_ppp_rcvdata_vjc_uncomp, 1503, 1509
- ng_ppp_rcvdata_vjc_vjip, 1504, 1509
- ng_ppp_rcvmsg, 1504, 1509
- ng_ppp_rseq_array_info, 1509
- ng_ppp_rseq_array_type, 1510
- ng_ppp_shutdown, 1504, 1510
- ng_ppp_start_frag_timer, 1505
- ng_ppp_stats_type, 1510
- ng_ppp_stats_type_fields, 1510
- ng_ppp_stop_frag_timer, 1505
- ng_ppp_tpestruct, 1510
- ng_ppp_update, 1505
- priv_p, 1492
- PROT_ATALK, 1491
- PROT_COMPD, 1491
- PROT_COMPRESSABLE, 1491
- PROT_CRYPTD, 1491
- PROT_IP, 1491
- PROT_IPV6, 1491
- PROT_IPX, 1492
- PROT_LCP, 1492
- PROT_MP, 1492
- PROT_VALID, 1492
- PROT_VJCOMP, 1492
- PROT_VJUNCOMP, 1492
- ng_ppp.h
 - NG_PPP_COMPRESS_FULL, 1515
 - NG_PPP_COMPRESS_NONE, 1515
 - NG_PPP_COMPRESS_SIMPLE, 1515
 - NG_PPP_DECOMPRESS_FULL, 1516
 - NG_PPP_DECOMPRESS_NONE, 1515
 - NG_PPP_DECOMPRESS_SIMPLE, 1515
 - NGM_PPP_CLR_LINK_STATS, 1516
 - NGM_PPP_GET_CONFIG, 1516
 - NGM_PPP_GET_LINK_STATS, 1516
 - NGM_PPP_GET_MP_STATE, 1516
 - NGM_PPP_GETCLR_LINK_STATS, 1516
 - NGM_PPP_SET_CONFIG, 1516

- ng_ppp.h
 - NG_PPP_BUND_TYPE_INFO, [1512](#)
 - NG_PPP_BUNDLE_LINKNUM, [1512](#)
 - NG_PPP_CONFIG_TYPE_INFO, [1512](#)
 - NG_PPP_HOOK_ATALK, [1512](#)
 - NG_PPP_HOOK_BYPASS, [1513](#)
 - NG_PPP_HOOK_COMPRESS, [1513](#)
 - NG_PPP_HOOK_DECOMPRESS, [1513](#)
 - NG_PPP_HOOK_DECRYPT, [1513](#)
 - NG_PPP_HOOK_ENCRYPT, [1513](#)
 - NG_PPP_HOOK_INET, [1513](#)
 - NG_PPP_HOOK_IPV6, [1513](#)
 - NG_PPP_HOOK_IPX, [1513](#)
 - NG_PPP_HOOK_LINK_PREFIX, [1513](#)
 - NG_PPP_HOOK_VJC_COMP, [1513](#)
 - NG_PPP_HOOK_VJC_IP, [1513](#)
 - NG_PPP_HOOK_VJC_UNCOMP, [1514](#)
 - NG_PPP_HOOK_VJC_VJIP, [1514](#)
 - NG_PPP_LINK_TYPE_INFO, [1514](#)
 - NG_PPP_MAX_BANDWIDTH, [1514](#)
 - NG_PPP_MAX_LATENCY, [1514](#)
 - NG_PPP_MAX_LINKS, [1514](#)
 - NG_PPP_MP_STATE_TYPE_INFO, [1514](#)
 - NG_PPP_NODE_TYPE, [1515](#)
 - NG_PPP_STATS_TYPE_INFO, [1515](#)
 - NGM_PPP_COOKIE, [1515](#)
- ng_ppp_acf
 - ng_ppp.c, [1506](#)
- ng_ppp_addproto
 - ng_ppp.c, [1493](#)
- ng_ppp_array_info
 - ng_ppp.c, [1506](#)
- ng_ppp_bund_conf, [352](#)
 - enableAtalk, [352](#)
 - enableCompression, [352](#)
 - enableDecompression, [352](#)
 - enableDecryption, [352](#)
 - enableEncryption, [352](#)
 - enableIP, [352](#)
 - enableIPv6, [353](#)
 - enableIPX, [353](#)
 - enableMultilink, [353](#)
 - enableRoundRobin, [353](#)
 - enableVJCompression, [353](#)
 - enableVJDecompression, [353](#)
 - mrru, [353](#)
 - recvShortSeq, [353](#)
 - xmitShortSeq, [353](#)
- ng_ppp_bund_type
 - ng_ppp.c, [1506](#)
- ng_ppp_bund_type_fields
 - ng_ppp.c, [1506](#)
- NG_PPP_BUND_TYPE_INFO
 - ng_ppp.h, [1512](#)
- NG_PPP_BUNDLE_LINKNUM
 - ng_ppp.h, [1512](#)
- ng_ppp_bypass
 - ng_ppp.c, [1493](#)
- ng_ppp_check_packet
 - ng_ppp.c, [1493](#)
- ng_ppp_cmds
 - ng_ppp.c, [1506](#)
- ng_ppp_comp_recv
 - ng_ppp.c, [1493](#)
- ng_ppp_comp_xmit
 - ng_ppp.c, [1494](#)
- NG_PPP_COMPRESS_FULL
 - ng_ppp.h, [1515](#)
- NG_PPP_COMPRESS_NONE
 - ng_ppp.h, [1515](#)
- NG_PPP_COMPRESS_SIMPLE
 - ng_ppp.h, [1515](#)
- ng_ppp_conf_type
 - ng_ppp.c, [1506](#)
- ng_ppp_conf_type_fields
 - ng_ppp.c, [1507](#)
- NG_PPP_CONFIG_TYPE_INFO
 - ng_ppp.h, [1512](#)
- ng_ppp_config_valid
 - ng_ppp.c, [1494](#)
- ng_ppp_constructor
 - ng_ppp.c, [1494](#), [1507](#)
- ng_ppp_crypt_recv
 - ng_ppp.c, [1494](#)
- ng_ppp_crypt_xmit
 - ng_ppp.c, [1495](#)
- ng_ppp_cutproto
 - ng_ppp.c, [1495](#)
- NG_PPP_DECOMPRESS_FULL
 - ng_ppp.h, [1516](#)
- NG_PPP_DECOMPRESS_NONE
 - ng_ppp.h, [1515](#)
- NG_PPP_DECOMPRESS_SIMPLE
 - ng_ppp.h, [1515](#)
- ng_ppp_disconnect
 - ng_ppp.c, [1495](#), [1507](#)
- ng_ppp_frag, [355](#)
 - data, [355](#)
 - first, [355](#)
 - last, [355](#)
 - seq, [355](#)
 - timestamp, [355](#)
- ng_ppp_frag_checkstale
 - ng_ppp.c, [1496](#)
- ng_ppp_frag_process
 - ng_ppp.c, [1496](#)
- ng_ppp_frag_reset
 - ng_ppp.c, [1496](#)

- ng_ppp_frag_timeout
 - ng_ppp.c, 1497
- ng_ppp_frag_trim
 - ng_ppp.c, 1497
- ng_ppp_get_packet
 - ng_ppp.c, 1497
- ng_ppp_hcomp_rcv
 - ng_ppp.c, 1497
- ng_ppp_hcomp_xmit
 - ng_ppp.c, 1498
- NG_PPP_HOOK_ATALK
 - ng_ppp.h, 1512
- NG_PPP_HOOK_BYPASS
 - ng_ppp.h, 1513
- NG_PPP_HOOK_COMPRESS
 - ng_ppp.h, 1513
- NG_PPP_HOOK_DECOMPRESS
 - ng_ppp.h, 1513
- NG_PPP_HOOK_DECRYPT
 - ng_ppp.h, 1513
- NG_PPP_HOOK_ENCRYPT
 - ng_ppp.h, 1513
- NG_PPP_HOOK_INET
 - ng_ppp.h, 1513
- NG_PPP_HOOK_IPV6
 - ng_ppp.h, 1513
- NG_PPP_HOOK_IPX
 - ng_ppp.h, 1513
- NG_PPP_HOOK_LINK_PREFIX
 - ng_ppp.h, 1513
- ng_ppp_hook_names
 - ng_ppp.c, 1507
- NG_PPP_HOOK_VJC_COMP
 - ng_ppp.h, 1513
- NG_PPP_HOOK_VJC_IP
 - ng_ppp.h, 1513
- NG_PPP_HOOK_VJC_UNCOMP
 - ng_ppp.h, 1514
- NG_PPP_HOOK_VJC_VJIP
 - ng_ppp.h, 1514
- ng_ppp_intcmp
 - ng_ppp.c, 1498
- ng_ppp_link, 357
 - bytesInQueue, 357
 - conf, 357
 - hook, 357
 - lastWrite, 358
 - latency, 358
 - seq, 358
 - stats, 358
- ng_ppp_link_array_type
 - ng_ppp.c, 1507
- ng_ppp_link_conf, 359
 - bandwidth, 359
 - enableACFComp, 359
 - enableLink, 359
 - enableProtoComp, 359
 - latency, 359
 - mru, 359
- ng_ppp_link_stat, 361
 - badProtos, 361
 - dropFragments, 361
 - dupFragments, 361
 - rcvFrames, 361
 - rcvOctets, 361
 - runts, 361
 - xmitFrames, 362
 - xmitOctets, 362
- ng_ppp_link_type
 - ng_ppp.c, 1507
- ng_ppp_link_type_fields
 - ng_ppp.c, 1507
- NG_PPP_LINK_TYPE_INFO
 - ng_ppp.h, 1514
- ng_ppp_link_xmit
 - ng_ppp.c, 1498
- NG_PPP_MAX_BANDWIDTH
 - ng_ppp.h, 1514
- NG_PPP_MAX_LATENCY
 - ng_ppp.h, 1514
- NG_PPP_MAX_LINKS
 - ng_ppp.h, 1514
- ng_ppp_max_staleness
 - ng_ppp.c, 1508
- ng_ppp_mp_rcv
 - ng_ppp.c, 1498
- ng_ppp_mp_state, 363
 - mseq, 363
 - rseq, 363
 - xseq, 363
- ng_ppp_mp_state_type
 - ng_ppp.c, 1508
- ng_ppp_mp_state_type_fields
 - ng_ppp.c, 1508
- NG_PPP_MP_STATE_TYPE_INFO
 - ng_ppp.h, 1514
- ng_ppp_mp_strategy
 - ng_ppp.c, 1499
- ng_ppp_mp_xmit
 - ng_ppp.c, 1499
- ng_ppp_newhook
 - ng_ppp.c, 1500, 1508
- ng_ppp_node_conf, 364
 - bund, 364
 - links, 364
- NG_PPP_NODE_TYPE
 - ng_ppp.h, 1515
- ng_ppp_prepend

- ng_ppp.c, 1500
- ng_ppp_private, 365
 - activeLinks, 365
 - allLinksEqual, 365
 - bundleStats, 365
 - conf, 365
 - hooks, 366
 - lastLink, 366
 - links, 366
 - mseq, 366
 - numActiveLinks, 366
 - vjCompHooked, 366
 - xseq, 366
- ng_ppp_proto_rcv
 - ng_ppp.c, 1500
- ng_ppp_rcvdata
 - ng_ppp.c, 1500, 1508
- ng_ppp_rcvdata_atalk
 - ng_ppp.c, 1501, 1508
- ng_ppp_rcvdata_bypass
 - ng_ppp.c, 1501, 1508
- ng_ppp_rcvdata_compress
 - ng_ppp.c, 1501, 1508
- ng_ppp_rcvdata_decompress
 - ng_ppp.c, 1501, 1508
- ng_ppp_rcvdata_decrypt
 - ng_ppp.c, 1502, 1508
- ng_ppp_rcvdata_encrypt
 - ng_ppp.c, 1502, 1509
- ng_ppp_rcvdata_inet
 - ng_ppp.c, 1502, 1509
- ng_ppp_rcvdata_ipv6
 - ng_ppp.c, 1503, 1509
- ng_ppp_rcvdata_ipx
 - ng_ppp.c, 1503, 1509
- ng_ppp_rcvdata_vjc_comp
 - ng_ppp.c, 1503, 1509
- ng_ppp_rcvdata_vjc_ip
 - ng_ppp.c, 1503, 1509
- ng_ppp_rcvdata_vjc_uncomp
 - ng_ppp.c, 1503, 1509
- ng_ppp_rcvdata_vjc_vjip
 - ng_ppp.c, 1504, 1509
- ng_ppp_rcvmsg
 - ng_ppp.c, 1504, 1509
- ng_ppp_rseq_array_info
 - ng_ppp.c, 1509
- ng_ppp_rseq_array_type
 - ng_ppp.c, 1510
- ng_ppp_shutdown
 - ng_ppp.c, 1504, 1510
- ng_ppp_start_frag_timer
 - ng_ppp.c, 1505
- ng_ppp_stats_type
 - ng_ppp.c, 1510
- ng_ppp_stats_type_fields
 - ng_ppp.c, 1510
- NG_PPP_STATS_TYPE_INFO
 - ng_ppp.h, 1515
- ng_ppp_stop_frag_timer
 - ng_ppp.c, 1505
- ng_ppp_tpestruct
 - ng_ppp.c, 1510
- ng_ppp_update
 - ng_ppp.c, 1505
- ng_pppoe.c
 - PPPOE_CONNECTED, 1520
 - PPPOE_DEAD, 1521
 - PPPOE_LISTENING, 1520
 - PPPOE_NEWCONNECTED, 1520
 - PPPOE_PRIMED, 1520
 - PPPOE_SINIT, 1520
 - PPPOE_SNONE, 1520
 - PPPOE_SOFFER, 1520
 - PPPOE_SREQ, 1520
- ng_pppoe.c
 - COMPAT_3COM, 1519
 - COMPAT_DLINK, 1519
 - get_new_sid, 1521
 - get_tag, 1521
 - init_tags, 1521
 - insert_tag, 1521
 - LEAVE, 1519
 - M_NETGRAPH_PPPOE, 1519
 - make_packet, 1521
 - negp, 1520
 - NETGRAPH_INIT, 1521
 - next_tag, 1522
 - ng_pppoe_cmds, 1528
 - ng_pppoe_connect, 1522, 1528
 - ng_pppoe_constructor, 1522, 1528
 - ng_pppoe_disconnect, 1522, 1528
 - ng_pppoe_newhook, 1522, 1528
 - ng_pppoe_rcvdata, 1522, 1528
 - ng_pppoe_rcvmsg, 1523, 1528
 - ng_pppoe_sendpacket, 1524
 - NG_PPPOE_SESSION_NODE, 1519
 - ng_pppoe_shutdown, 1525, 1528
 - ng_pppoe_sts_state_type, 1528
 - ng_pppoe_sts_type_fields, 1528
 - ngpppoe_init_data_state_type, 1528
 - ngpppoe_init_data_type_fields, 1529
 - NUMTAGS, 1520
 - OFFSETOF, 1520
 - pppoe_broadcast_padi, 1525
 - pppoe_find_svc, 1525
 - pppoe_findsession, 1525
 - pppoe_finduniq, 1525

- pppoe_match_svc, 1525
- pppoe_send_event, 1525
- pppoe_start, 1526
- pppoe_ticker, 1526
- priv_p, 1520
- scan_tags, 1527
- send_acname, 1527
- send_sessionid, 1527
- sessp, 1520
- SIGNOFF, 1520
- state, 1520
- typestruct, 1529
- ng_pppoe.h
 - NGM_PPPOE_ACNAME, 1537
 - NGM_PPPOE_CLOSE, 1537
 - NGM_PPPOE_CONNECT, 1537
 - NGM_PPPOE_FAIL, 1537
 - NGM_PPPOE_GET_STATUS, 1537
 - NGM_PPPOE_GETMODE, 1537
 - NGM_PPPOE_LISTEN, 1537
 - NGM_PPPOE_OFFER, 1537
 - NGM_PPPOE_SERVICE, 1537
 - NGM_PPPOE_SESSIONID, 1537
 - NGM_PPPOE_SET_FLAG, 1537
 - NGM_PPPOE_SETENADDR, 1537
 - NGM_PPPOE_SETMODE, 1537
 - NGM_PPPOE_SUCCESS, 1537
- ng_pppoe.h
 - __packed, 1538
 - cmd, 1537
 - ETHERTYPE_PPPOE_3COM_DISC, 1531
 - ETHERTYPE_PPPOE_3COM_SESS, 1531
 - ETHERTYPE_PPPOE_DISC, 1532
 - ETHERTYPE_PPPOE_SESS, 1532
 - NG_PPPOE_3COM, 1532
 - NG_PPPOE_DLINK, 1532
 - NG_PPPOE_HOOK_C_LEADIN, 1532
 - NG_PPPOE_HOOK_DEBUG, 1532
 - NG_PPPOE_HOOK_ETHERNET, 1532
 - NG_PPPOE_HOOK_PADI, 1532
 - NG_PPPOE_HOOK_S_LEADIN, 1532
 - NG_PPPOE_INIT_DATA_TYPE_INFO, 1533
 - NG_PPPOE_NODE_TYPE, 1533
 - NG_PPPOE_NONSTANDARD, 1533
 - NG_PPPOE_STANDARD, 1533
 - NG_PPPOE_STS_TYPE_INFO, 1533
 - NG_PPPOESTAT_TYPE_INFO, 1533
 - NGM_PPPOE_COOKIE, 1533
 - PADI_CODE, 1534
 - PADO_CODE, 1534
 - PADR_CODE, 1534
 - PADS_CODE, 1534
 - PADT_CODE, 1534
 - PPPOE_INITIAL_TIMEOUT, 1534
 - PPPOE_NUM_SESSIONS, 1534
 - PPPOE_OFFER_TIMEOUT, 1534
 - PPPOE_SERVICE_NAME_SIZE, 1534
 - PPPOE_TIMEOUT_LIMIT, 1535
 - PTT_AC_COOKIE, 1535
 - PTT_AC_NAME, 1535
 - PTT_EOL, 1535
 - PTT_GEN_ERR, 1535
 - PTT_HOST_UNIQ, 1535
 - PTT_RELAY_SID, 1535
 - PTT_SRV_ERR, 1535
 - PTT_SRV_NAME, 1535
 - PTT_SYS_ERR, 1536
 - PTT_VENDOR, 1536
 - TAGI_HUNIQ, 1536
 - TAGI_SVC, 1536
 - TAGO_ACNAME, 1536
 - TAGO_COOKIE, 1536
 - TAGO_HUNIQ, 1536
 - TAGO_SVC, 1536
 - TAGR_COOKIE, 1536
 - TAGR_HUNIQ, 1536
 - TAGR_SVC, 1537
 - TAGS_ACNAME, 1537
 - TAGS_COOKIE, 1537
 - TAGS_HUNIQ, 1537
 - TAGS_SVC, 1537
 - NG_PPPOE_3COM
 - ng_pppoe.h, 1532
 - ng_pppoe_cmds
 - ng_pppoe.c, 1528
 - ng_pppoe_connect
 - ng_pppoe.c, 1522, 1528
 - ng_pppoe_constructor
 - ng_pppoe.c, 1522, 1528
 - ng_pppoe_disconnect
 - ng_pppoe.c, 1522, 1528
 - NG_PPPOE_DLINK
 - ng_pppoe.h, 1532
 - NG_PPPOE_HOOK_C_LEADIN
 - ng_pppoe.h, 1532
 - NG_PPPOE_HOOK_DEBUG
 - ng_pppoe.h, 1532
 - NG_PPPOE_HOOK_ETHERNET
 - ng_pppoe.h, 1532
 - NG_PPPOE_HOOK_PADI
 - ng_pppoe.h, 1532
 - NG_PPPOE_HOOK_S_LEADIN
 - ng_pppoe.h, 1532
 - NG_PPPOE_INIT_DATA_TYPE_INFO
 - ng_pppoe.h, 1533
 - ng_pppoe_newhook
 - ng_pppoe.c, 1522, 1528

- NG_PPPOE_NODE_TYPE
 - ng_pppoe.h, 1533
- NG_PPPOE_NONSTANDARD
 - ng_pppoe.h, 1533
- ng_pppoe_rcvdata
 - ng_pppoe.c, 1522, 1528
- ng_pppoe_rcvmgs
 - ng_pppoe.c, 1523, 1528
- ng_pppoe_sendpacket
 - ng_pppoe.c, 1524
- NG_PPPOE_SESSION_NODE
 - ng_pppoe.c, 1519
- ng_pppoe_shutdown
 - ng_pppoe.c, 1525, 1528
- NG_PPPOE_STANDARD
 - ng_pppoe.h, 1533
- ng_pppoe_sts_state_type
 - ng_pppoe.c, 1528
- ng_pppoe_sts_type_fields
 - ng_pppoe.c, 1528
- NG_PPPOE_STSTYPE_INFO
 - ng_pppoe.h, 1533
- NG_PPPOESTAT_TYPE_INFO
 - ng_pppoe.h, 1533
- ng_pptp_stats_type
 - ng_pptpgre.c, 1549
- ng_pptpgre.c
 - ERROUT, 1542
 - NETGRAPH_INIT, 1544
 - ng_pptp_stats_type, 1549
 - ng_pptpgre_cmdlist, 1549
 - ng_pptpgre_conf_type, 1549
 - ng_pptpgre_conf_type_fields, 1549
 - ng_pptpgre_constructor, 1544, 1549
 - ng_pptpgre_disconnect, 1544, 1549
 - ng_pptpgre_newhook, 1544, 1549
 - ng_pptpgre_rcvdata, 1544, 1549
 - ng_pptpgre_rcvmgs, 1545, 1549
 - ng_pptpgre_recv, 1545
 - ng_pptpgre_recv_ack_timeout, 1545
 - ng_pptpgre_reset, 1546
 - ng_pptpgre_send_ack_timeout, 1546
 - ng_pptpgre_shutdown, 1546, 1550
 - ng_pptpgre_start_recv_ack_timer, 1547
 - ng_pptpgre_start_send_ack_timer, 1547
 - ng_pptpgre_stats_type_fields, 1550
 - ng_pptpgre_stop_recv_ack_timer, 1547
 - ng_pptpgre_stop_send_ack_timer, 1548
 - ng_pptpgre_time, 1548
 - ng_pptpgre_tpestruct, 1550
 - ng_pptpgre_xmit, 1548
 - PPTP_ACK_ALPHA, 1542
 - PPTP_ACK_BETA, 1542
 - PPTP_ACK_CHI, 1542
 - PPTP_ACK_DELTA, 1542
 - PPTP_GRE_PROTO, 1542
 - PPTP_INIT_MASK, 1542
 - PPTP_INIT_VALUE, 1542
 - PPTP_MAX_ACK_DELAY, 1542
 - PPTP_MAX_PAYLOAD, 1542
 - PPTP_MAX_TIMEOUT, 1543
 - PPTP_MIN_ACK_DELAY, 1543
 - PPTP_MIN_RTT, 1543
 - PPTP_MIN_TIMEOUT, 1543
 - PPTP_SEQ_DIFF, 1543
 - PPTP_TIME_SCALE, 1543
 - PPTP_XMIT_WIN, 1543
 - pptptime_t, 1544
 - priv_p, 1544
- ng_pptpgre.h
 - NGM_PPTPGRE_CLR_STATS, 1553
 - NGM_PPTPGRE_GET_CONFIG, 1552
 - NGM_PPTPGRE_GET_STATS, 1553
 - NGM_PPTPGRE_GETCLR_STATS, 1553
 - NGM_PPTPGRE_SET_CONFIG, 1552
- ng_pptpgre.h
 - NG_PPTPGRE_CONF_TYPE_INFO, 1551
 - NG_PPTPGRE_HOOK_LOWER, 1551
 - NG_PPTPGRE_HOOK_UPPER, 1552
 - NG_PPTPGRE_NODE_TYPE, 1552
 - NG_PPTPGRE_STATS_TYPE_INFO, 1552
 - NGM_PPTPGRE_COOKIE, 1552
- ng_pptpgre_ackp, 367
 - ato, 367
 - dev, 367
 - rackTimer, 367
 - rtt, 367
 - sackTimer, 367
 - timeSent, 367
 - winAck, 368
 - xmitWin, 368
- ng_pptpgre_cmdlist
 - ng_pptpgre.c, 1549
- ng_pptpgre_conf, 369
 - cid, 369
 - enableAlwaysAck, 369
 - enabled, 369
 - enableDelayedAck, 369
 - enableWindowing, 369
 - peerCid, 369
 - peerPpd, 369
 - recvWin, 370
- ng_pptpgre_conf_type
 - ng_pptpgre.c, 1549
- ng_pptpgre_conf_type_fields
 - ng_pptpgre.c, 1549
- NG_PPTPGRE_CONF_TYPE_INFO
 - ng_pptpgre.h, 1551

- ng_pptpgre_constructor
 - ng_pptpgre.c, 1544, 1549
- ng_pptpgre_disconnect
 - ng_pptpgre.c, 1544, 1549
- NG_PPTPGRE_HOOK_LOWER
 - ng_pptpgre.h, 1551
- NG_PPTPGRE_HOOK_UPPER
 - ng_pptpgre.h, 1552
- ng_pptpgre_newhook
 - ng_pptpgre.c, 1544, 1549
- NG_PPTPGRE_NODE_TYPE
 - ng_pptpgre.h, 1552
- ng_pptpgre_private, 371
 - ackp, 371
 - conf, 371
 - lower, 371
 - mtx, 372
 - recvAck, 372
 - recvSeq, 372
 - startTime, 372
 - stats, 372
 - upper, 372
 - xmitAck, 372
 - xmitSeq, 372
- ng_pptpgre_rcvdata
 - ng_pptpgre.c, 1544, 1549
- ng_pptpgre_rcvmsg
 - ng_pptpgre.c, 1545, 1549
- ng_pptpgre_recv
 - ng_pptpgre.c, 1545
- ng_pptpgre_recv_ack_timeout
 - ng_pptpgre.c, 1545
- ng_pptpgre_reset
 - ng_pptpgre.c, 1546
- ng_pptpgre_send_ack_timeout
 - ng_pptpgre.c, 1546
- ng_pptpgre_shutdown
 - ng_pptpgre.c, 1546, 1550
- ng_pptpgre_start_recv_ack_timer
 - ng_pptpgre.c, 1547
- ng_pptpgre_start_send_ack_timer
 - ng_pptpgre.c, 1547
- ng_pptpgre_stats, 373
 - memoryFailures, 373
 - recvAckTimeouts, 373
 - recvBadAcks, 373
 - recvBadCID, 373
 - recvBadGRE, 373
 - recvDuplicates, 373
 - recvLoneAcks, 374
 - recvOctets, 374
 - recvOutOfOrder, 374
 - recvPackets, 374
 - recvRunts, 374
 - xmitDrops, 374
 - xmitLoneAcks, 374
 - xmitOctets, 374
 - xmitPackets, 374
 - xmitTooBig, 374
- ng_pptpgre_stats_type_fields
 - ng_pptpgre.c, 1550
- NG_PPTPGRE_STATS_TYPE_INFO
 - ng_pptpgre.h, 1552
- ng_pptpgre_stop_recv_ack_timer
 - ng_pptpgre.c, 1547
- ng_pptpgre_stop_send_ack_timer
 - ng_pptpgre.c, 1548
- ng_pptpgre_time
 - ng_pptpgre.c, 1548
- ng_pptpgre_typestruct
 - ng_pptpgre.c, 1550
- ng_pptpgre_xmit
 - ng_pptpgre.c, 1548
- ng_pred1.c
 - Crc16, 1556
 - Crc16Table, 1559
 - ERROUT, 1555
 - HASH, 1555
 - MALLOC_DEFINE, 1556
 - NETGRAPH_INIT, 1557
 - ng_pred1_cmds, 1559
 - ng_pred1_compress, 1557
 - ng_pred1_config_type, 1559
 - ng_pred1_config_type_fields, 1560
 - ng_pred1_constructor, 1557, 1560
 - ng_pred1_decompress, 1557
 - ng_pred1_disconnect, 1557, 1560
 - ng_pred1_newhook, 1558, 1560
 - ng_pred1_rcvdata, 1558, 1560
 - ng_pred1_rcvmsg, 1558, 1560
 - ng_pred1_shutdown, 1558, 1560
 - ng_pred1_stat_type, 1560
 - ng_pred1_stats_type_fields, 1560
 - ng_pred1_typestruct, 1561
 - PPP_GOODFCS, 1556
 - PPP_INITFCS, 1556
 - PRED1_BUF_SIZE, 1556
 - PRED1_HDRLEN, 1556
 - PRED1_TABLE_SIZE, 1556
 - Pred1Compress, 1559
 - Pred1Decompress, 1559
 - Pred1Init, 1559
 - Pred1SyncTable, 1559
 - priv_p, 1556
- ng_pred1.h
 - NGM_PRED1_CLR_STATS, 1563
 - NGM_PRED1_CONFIG, 1563
 - NGM_PRED1_GET_STATS, 1563

- NGM_PRED1_GETCLR_STATS, 1563
- NGM_PRED1_RESETRREQ, 1563
- ng_pred1.h
 - NG_PRED1_CONFIG_INFO, 1562
 - NG_PRED1_HOOK_COMP, 1562
 - NG_PRED1_HOOK_DECOMP, 1562
 - NG_PRED1_NODE_TYPE, 1563
 - NG_PRED1_STATS_INFO, 1563
 - NGM_PRED1_COOKIE, 1563
- ng_pred1_cmds
 - ng_pred1.c, 1559
- ng_pred1_compress
 - ng_pred1.c, 1557
- ng_pred1_config, 375
 - enable, 375
- NG_PRED1_CONFIG_INFO
 - ng_pred1.h, 1562
- ng_pred1_config_type
 - ng_pred1.c, 1559
- ng_pred1_config_type_fields
 - ng_pred1.c, 1560
- ng_pred1_constructor
 - ng_pred1.c, 1557, 1560
- ng_pred1_decompress
 - ng_pred1.c, 1557
- ng_pred1_disconnect
 - ng_pred1.c, 1557, 1560
- NG_PRED1_HOOK_COMP
 - ng_pred1.h, 1562
- NG_PRED1_HOOK_DECOMP
 - ng_pred1.h, 1562
- ng_pred1_newhook
 - ng_pred1.c, 1558, 1560
- NG_PRED1_NODE_TYPE
 - ng_pred1.h, 1563
- ng_pred1_private, 376
 - cfg, 376
 - compress, 376
 - ctrlnode, 376
 - GuessTable, 376
 - Hash, 376
 - inbuf, 376
 - outbuf, 377
 - seqnum, 377
 - stats, 377
- ng_pred1_rcvdata
 - ng_pred1.c, 1558, 1560
- ng_pred1_rcvmsg
 - ng_pred1.c, 1558, 1560
- ng_pred1_shutdown
 - ng_pred1.c, 1558, 1560
- ng_pred1_stat_type
 - ng_pred1.c, 1560
- ng_pred1_stats, 378
 - Errors, 378
 - FramesComp, 378
 - FramesPlain, 378
 - FramesUncomp, 378
 - InOctets, 378
 - OutOctets, 378
- NG_PRED1_STATS_INFO
 - ng_pred1.h, 1563
- ng_pred1_stats_type_fields
 - ng_pred1.c, 1560
- ng_pred1_typestruct
 - ng_pred1.c, 1561
- NG_PRIO_CUTOFF
 - netgraph.h, 1140
- NG_PRIO_LINKSTATE
 - netgraph.h, 1140
- NG_PROGRESS
 - netgraph.h, 1140
- NG_QUEUE
 - netgraph.h, 1140
- ng_queue, 379
 - last, 379
 - q_flags, 379
 - q_mtx, 380
 - q_node, 380
 - queue, 380
- ng_queue_rw
 - ng_base.c, 1195
- ng_qzone
 - ng_base.c, 1203
- ng_rcvdata_t
 - netgraph.h, 1151
- ng_rcvitem
 - netgraph.h, 1151
- ng_rcvmsg_t
 - netgraph.h, 1151
- NG_REALLY_DIE
 - netgraph.h, 1140
- ng_replace_retaddr
 - netgraph.h, 1155
 - ng_base.c, 1195
- NG_RESPOND_MSG
 - netgraph.h, 1140
- ng_rfc1490.c
 - NG_RFC1490_ENCAP_CISCO, 1567
 - NG_RFC1490_ENCAP_IETF_IP, 1567
 - NG_RFC1490_ENCAP_IETF_SNAP, 1567
- ng_rfc1490.c
 - ERROUT, 1566
 - HDLC_UI, 1566
 - MAX_ENCAPS_HDR, 1566
 - NETGRAPH_INIT, 1567
 - ng_rfc1490_cmds, 1568
 - ng_rfc1490_constructor, 1567, 1569

- ng_rfc1490_disconnect, [1567](#), [1569](#)
- ng_rfc1490_encaps, [1569](#)
- ng_rfc1490_newhook, [1568](#), [1569](#)
- ng_rfc1490_rcvdata, [1568](#), [1569](#)
- ng_rfc1490_rcvmsg, [1568](#), [1569](#)
- ng_rfc1490_shutdown, [1568](#), [1569](#)
- NLPID_CLNP, [1566](#)
- NLPID_ESIS, [1566](#)
- NLPID_IP, [1566](#)
- NLPID_ISIS, [1566](#)
- NLPID_PPP, [1566](#)
- NLPID_Q933, [1566](#)
- NLPID_SNAP, [1567](#)
- OUICMP, [1567](#)
- priv_p, [1567](#)
- typestruct, [1569](#)
- ng_rfc1490.h
 - NGM_RFC1490_GET_ENCAP, [1572](#)
 - NGM_RFC1490_SET_ENCAP, [1572](#)
- ng_rfc1490.h
 - NG_RFC1490_HOOK_DOWNSTREAM, [1571](#)
 - NG_RFC1490_HOOK_ETHERNET, [1571](#)
 - NG_RFC1490_HOOK_INET, [1571](#)
 - NG_RFC1490_HOOK_PPP, [1571](#)
 - NG_RFC1490_NODE_TYPE, [1571](#)
 - NGM_RFC1490_COOKIE, [1571](#)
- ng_rfc1490_cmds
 - ng_rfc1490.c, [1568](#)
- ng_rfc1490_constructor
 - ng_rfc1490.c, [1567](#), [1569](#)
- ng_rfc1490_disconnect
 - ng_rfc1490.c, [1567](#), [1569](#)
- NG_RFC1490_ENCAP_CISCO
 - ng_rfc1490.c, [1567](#)
- NG_RFC1490_ENCAP_IETF_IP
 - ng_rfc1490.c, [1567](#)
- NG_RFC1490_ENCAP_IETF_SNAP
 - ng_rfc1490.c, [1567](#)
- ng_rfc1490_encap_t, [381](#)
 - method, [381](#)
 - name, [381](#)
- ng_rfc1490_encaps
 - ng_rfc1490.c, [1569](#)
- NG_RFC1490_HOOK_DOWNSTREAM
 - ng_rfc1490.h, [1571](#)
- NG_RFC1490_HOOK_ETHERNET
 - ng_rfc1490.h, [1571](#)
- NG_RFC1490_HOOK_INET
 - ng_rfc1490.h, [1571](#)
- NG_RFC1490_HOOK_PPP
 - ng_rfc1490.h, [1571](#)
- ng_rfc1490_newhook
 - ng_rfc1490.c, [1568](#), [1569](#)
- NG_RFC1490_NODE_TYPE
 - ng_rfc1490.h, [1571](#)
- ng_rfc1490_private, [382](#)
 - downlink, [382](#)
 - enc, [382](#)
 - ethernet, [382](#)
 - inet, [382](#)
 - ppp, [383](#)
- ng_rfc1490_rcvdata
 - ng_rfc1490.c, [1568](#), [1569](#)
- ng_rfc1490_rcvmsg
 - ng_rfc1490.c, [1568](#), [1569](#)
- ng_rfc1490_shutdown
 - ng_rfc1490.c, [1568](#), [1569](#)
- ng_rmhook_part2
 - ng_base.c, [1195](#)
- ng_rmhook_self
 - netgraph.h, [1155](#)
 - ng_base.c, [1195](#)
- ng_rmnnode
 - ng_base.c, [1195](#)
- ng_rmnnode_self
 - netgraph.h, [1155](#)
 - ng_base.c, [1196](#)
- ng_rmtime
 - netgraph.h, [1156](#)
 - ng_base.c, [1196](#)
- ng_rx_iqfull
 - hva_stats_ng, [93](#)
- ng_rx_novcc
 - hva_stats_ng, [93](#)
- ng_rx_pdu
 - hva_stats_ng, [93](#)
- ng_rx_rawcell
 - hva_stats_ng, [93](#)
- ng_sample.c
 - M_NETGRAPH_XXX, [1574](#)
 - NETGRAPH_INIT, [1575](#)
 - ng_XXX_cmdlist, [1576](#)
 - ng_XXX_connect, [1575](#), [1576](#)
 - ng_XXX_constructor, [1575](#), [1576](#)
 - ng_XXX_disconnect, [1575](#), [1576](#)
 - ng_XXX_newhook, [1575](#), [1576](#)
 - ng_XXX_rcvdata, [1575](#), [1577](#)
 - ng_XXX_rcvmsg, [1575](#), [1577](#)
 - ng_XXX_shutdown, [1576](#), [1577](#)
 - ng_XXX_stat_type, [1577](#)
 - ng_XXX_stat_type_fields, [1577](#)
 - typestruct, [1577](#)
 - xxx_p, [1574](#)
- ng_sample.h
 - NGM_XXX_GET_STATUS, [1579](#)
 - NGM_XXX_SET_FLAG, [1579](#)
- ng_sample.h

- NG_XXX_HOOK_DEBUG, 1578
- NG_XXX_HOOK_DLCI_LEADIN, 1578
- NG_XXX_HOOK_DOWNSTREAM, 1578
- NG_XXX_NODE_TYPE, 1578
- NG_XXX_STATS_TYPE_INFO, 1578
- NGM_XXX_COOKIE, 1579
- XXX_NUM_DLCIS, 1579
- NG_SEND_DATA
 - netgraph.h, 1141
- NG_SEND_DATA_FLAGS
 - netgraph.h, 1141
- NG_SEND_DATA_ONLY
 - netgraph.h, 1141
- ng_send_fn
 - netgraph.h, 1142
- ng_send_fn1
 - netgraph.h, 1156
 - ng_base.c, 1197
- NG_SEND_MSG_HOOK
 - netgraph.h, 1142
- NG_SEND_MSG_ID
 - netgraph.h, 1142
- NG_SEND_MSG_PATH
 - netgraph.h, 1143
- NG_SEPARATE_MALLOC
 - netgraph.h, 1143
- ng_setisr
 - ng_base.c, 1197
- ng_setsockaddr
 - ng_socket.c, 1585
- ng_shutdown_t
 - netgraph.h, 1151
- ng_sizedstring_getDefault
 - ng_parse.c, 1463
- ng_sizedstring_parse
 - ng_parse.c, 1463
- ng_sizedstring_unparse
 - ng_parse.c, 1463
- ng_snd_item
 - netgraph.h, 1156
 - ng_base.c, 1197
- ng_socket
 - ngpcb, 450
- ng_socket.c
 - dummy_disconnect, 1583
 - M_NETGRAPH_PATH, 1583
 - M_NETGRAPH_SOCKET, 1583
 - NETGRAPH_INIT_ORDERED, 1583
 - ng_attach_cntl, 1583
 - ng_attach_common, 1583
 - ng_attach_data, 1584
 - ng_bind, 1584
 - ng_connect_data, 1584
 - ng_detach_common, 1584
 - ng_setsockaddr, 1585
 - ng_socket_free_priv, 1585
 - ng_socket_item_applied, 1585
 - ngc_attach, 1585
 - ngc_bind, 1586
 - ngc_connect, 1586
 - ngc_detach, 1586
 - ngc_send, 1586
 - ngc_usrreqs, 1590
 - ngd_attach, 1587
 - ngd_connect, 1587
 - ngd_detach, 1587
 - ngd_send, 1588
 - ngd_usrreqs, 1590
 - ngdomain, 1591
 - ngpdg_recvspace, 1591
 - ngpdg_sendspace, 1591
 - ngs_connect, 1588, 1591
 - ngs_constructor, 1588, 1591
 - ngs_disconnect, 1588, 1591
 - ngs_mod_event, 1588
 - ngs_newhook, 1588, 1591
 - ngs_rcvdata, 1589, 1591
 - ngs_rcvmsg, 1589, 1591
 - ngs_shutdown, 1589, 1592
 - ngsw, 1592
 - sotongpcb, 1583
 - SYSCTL_INT, 1589, 1590
 - SYSCTL_NODE, 1590
 - typestruct, 1592
- ng_socket.h
 - NGM_SOCKET_CMD_LINGER, 1594
 - NGM_SOCKET_CMD_NOLINGER, 1594
- ng_socket.h
 - NG_CONTROL, 1593
 - NG_DATA, 1593
 - NG_SOCKET_NODE_TYPE, 1593
 - NGM_SOCKET_COOKIE, 1593
- ng_socket_free_priv
 - ng_socket.c, 1585
- ng_socket_item_applied
 - ng_socket.c, 1585
- NG_SOCKET_NODE_TYPE
 - ng_socket.h, 1593
- ng_socketvar.h
 - NGS_FLAG_NOLINGER, 1595
- ng_source.c
 - __FBSDID, 1599
 - NETGRAPH_INIT, 1599
 - NG_SOURCE_ACTIVE, 1598
 - ng_source_clr_data, 1599
 - ng_source_cmds, 1602
 - ng_source_connect, 1599, 1602
 - ng_source_constructor, 1599, 1602

- ng_source_disconnect, 1599, 1602
- NG_SOURCE_DRIVER_IFQ_MAXLEN, 1598
- ng_source_intr, 1599
- NG_SOURCE_INTR_TICKS, 1598
- ng_source_newhook, 1600, 1602
- ng_source_rcvdata, 1600, 1602
- ng_source_rcvmsg, 1600, 1603
- ng_source_rmnode, 1601, 1603
- ng_source_send, 1601
- ng_source_set_autosrc, 1601
- ng_source_start, 1601
- ng_source_stats_type, 1603
- ng_source_stats_type_fields, 1603
- ng_source_stop, 1602
- ng_source_store_output_ifp, 1602
- ng_source_timeval_type, 1603
- ng_source_timeval_type_fields, 1603
- ng_source_tpestruct, 1603
- sc_p, 1599
- ng_source.h
 - NGM_SOURCE_CLR_DATA, 1606
 - NGM_SOURCE_CLR_STATS, 1606
 - NGM_SOURCE_GET_STATS, 1606
 - NGM_SOURCE_GETCLR_STATS, 1606
 - NGM_SOURCE_SETIFACE, 1606
 - NGM_SOURCE_SETPPS, 1606
 - NGM_SOURCE_START, 1606
 - NGM_SOURCE_STOP, 1606
- ng_source.h
 - NG_SOURCE_HOOK_INPUT, 1605
 - NG_SOURCE_HOOK_OUTPUT, 1605
 - NG_SOURCE_NODE_TYPE, 1605
 - NG_SOURCE_STATS_TYPE_INFO, 1605
 - ng_source_timeval_type, 1606
 - NGM_SOURCE_COOKIE, 1606
- NG_SOURCE_ACTIVE
 - ng_source.c, 1598
- ng_source_clr_data
 - ng_source.c, 1599
- ng_source_cmds
 - ng_source.c, 1602
- ng_source_connect
 - ng_source.c, 1599, 1602
- ng_source_constructor
 - ng_source.c, 1599, 1602
- ng_source_disconnect
 - ng_source.c, 1599, 1602
- NG_SOURCE_DRIVER_IFQ_MAXLEN
 - ng_source.c, 1598
- NG_SOURCE_HOOK_INPUT
 - ng_source.h, 1605
- NG_SOURCE_HOOK_OUTPUT
 - ng_source.h, 1605
- ng_source_intr
 - ng_source.c, 1599
- NG_SOURCE_INTR_TICKS
 - ng_source.c, 1598
- ng_source_newhook
 - ng_source.c, 1600, 1602
- NG_SOURCE_NODE_TYPE
 - ng_source.h, 1605
- ng_source_rcvdata
 - ng_source.c, 1600, 1602
- ng_source_rcvmsg
 - ng_source.c, 1600, 1603
- ng_source_rmnode
 - ng_source.c, 1601, 1603
- ng_source_send
 - ng_source.c, 1601
- ng_source_set_autosrc
 - ng_source.c, 1601
- ng_source_start
 - ng_source.c, 1601
- ng_source_stats, 384
 - elapsedTime, 384
 - endTime, 384
 - lastTime, 384
 - maxPps, 384
 - outFrames, 384
 - outOctets, 384
 - queueFrames, 384
 - queueOctets, 385
 - startTime, 385
- ng_source_stats_type
 - ng_source.c, 1603
- ng_source_stats_type_fields
 - ng_source.c, 1603
- NG_SOURCE_STATS_TYPE_INFO
 - ng_source.h, 1605
- ng_source_stop
 - ng_source.c, 1602
- ng_source_store_output_ifp
 - ng_source.c, 1602
- ng_source_timeval_type
 - ng_source.c, 1603
 - ng_source.h, 1606
- ng_source_timeval_type_fields
 - ng_source.c, 1603
- ng_source_tpestruct
 - ng_source.c, 1603
- ng_split.c
 - NETGRAPH_INIT, 1608
 - ng_split_constructor, 1608, 1609
 - ng_split_disconnect, 1608, 1609
 - ng_split_newhook, 1608, 1609
 - ng_split_rcvdata, 1609
 - ng_split_shutdown, 1609

- priv_p, 1608
- typestruct, 1609
- ng_split.h
 - NG_SPLIT_HOOK_IN, 1611
 - NG_SPLIT_HOOK_MIXED, 1611
 - NG_SPLIT_HOOK_OUT, 1611
 - NG_SPLIT_NODE_TYPE, 1611
 - NGM_SPLIT_COOKIE, 1611
- ng_split_constructor
 - ng_split.c, 1608, 1609
- ng_split_disconnect
 - ng_split.c, 1608, 1609
- NG_SPLIT_HOOK_IN
 - ng_split.h, 1611
- NG_SPLIT_HOOK_MIXED
 - ng_split.h, 1611
- NG_SPLIT_HOOK_OUT
 - ng_split.h, 1611
- ng_split_newhook
 - ng_split.c, 1608, 1609
- NG_SPLIT_NODE_TYPE
 - ng_split.h, 1611
- ng_split_private, 386
 - in, 386
 - mixed, 386
 - node, 386
 - out, 387
- ng_split_rcvdata
 - ng_split.c, 1609
- ng_split_shutdown
 - ng_split.c, 1609
- ng_sppp.c
 - __FBSDID, 1615
 - M_NETGRAPH_SPPP, 1614
 - MODULE_DEPEND, 1615
 - NETGRAPH_INIT, 1615
 - ng_sppp_cmds, 1616
 - ng_sppp_constructor, 1615, 1617
 - ng_sppp_disconnect, 1615, 1617
 - ng_sppp_free_unit, 1615
 - ng_sppp_get_unit, 1615
 - ng_sppp_ioctl, 1615
 - ng_sppp_newhook, 1615, 1617
 - ng_sppp_rcvdata, 1616, 1617
 - ng_sppp_rcvmsg, 1616, 1617
 - ng_sppp_shutdown, 1616, 1617
 - ng_sppp_start, 1616
 - ng_sppp_units, 1617
 - ng_sppp_units_len, 1617
 - ng_units_in_use, 1617
 - priv_p, 1614
 - typestruct, 1617
- ng_sppp.h
 - NGM_SPPP_GET_IFNAME, 1619
- ng_sppp.h
 - NG_SPPP_HOOK_DOWNSTREAM, 1619
 - NG_SPPP_IFACE_NAME, 1619
 - NG_SPPP_NODE_TYPE, 1619
 - NGM_SPPP_COOKIE, 1619
- ng_sppp_cmds
 - ng_sppp.c, 1616
- ng_sppp_constructor
 - ng_sppp.c, 1615, 1617
- ng_sppp_disconnect
 - ng_sppp.c, 1615, 1617
- ng_sppp_free_unit
 - ng_sppp.c, 1615
- ng_sppp_get_unit
 - ng_sppp.c, 1615
- NG_SPPP_HOOK_DOWNSTREAM
 - ng_sppp.h, 1619
- NG_SPPP_IFACE_NAME
 - ng_sppp.h, 1619
- ng_sppp_ioctl
 - ng_sppp.c, 1615
- ng_sppp_newhook
 - ng_sppp.c, 1615, 1617
- NG_SPPP_NODE_TYPE
 - ng_sppp.h, 1619
- ng_sppp_private, 388
 - hook, 388
 - ifp, 388
 - node, 388
 - unit, 389
- ng_sppp_rcvdata
 - ng_sppp.c, 1616, 1617
- ng_sppp_rcvmsg
 - ng_sppp.c, 1616, 1617
- ng_sppp_shutdown
 - ng_sppp.c, 1616, 1617
- ng_sppp_start
 - ng_sppp.c, 1616
- ng_sppp_units
 - ng_sppp.c, 1617
- ng_sppp_units_len
 - ng_sppp.c, 1617
- ng_sscfu.c
 - __FBSDID, 613
 - MALLOC_DEFINE, 613
 - MODULE_DEPEND, 613
 - NETGRAPH_INIT, 613
 - ng_sscfu_cmdlist, 616
 - ng_sscfu_constructor, 613, 616
 - ng_sscfu_disconnect, 613, 616
 - ng_sscfu_getdefparam_type, 616
 - ng_sscfu_getdefparam_type_info, 616
 - ng_sscfu_mod_event, 614
 - ng_sscfu_newhook, 614, 616

- ng_sscfu_rcvlower, 614, 616
- ng_sscfu_rcvmsg, 614, 616
- ng_sscfu_rcvupper, 614, 616
- ng_sscfu_shutdown, 614, 616
- ng_sscfu_tpestruct, 617
- ng_sscop_param_type, 617
- ng_sscop_param_type_info, 617
- sscfu_send_lower, 615
- sscfu_send_upper, 615
- sscfu_verbose, 615
- sscfu_window, 615
- text_status, 615
- ng_sscfu.h
 - NGM_SSCFU_DISABLE, 598
 - NGM_SSCFU_ENABLE, 598
 - NGM_SSCFU_GETDEBUG, 598
 - NGM_SSCFU_GETDEFPARAM, 598
 - NGM_SSCFU_GETSTATE, 598
 - NGM_SSCFU_SETDEBUG, 598
- ng_sscfu.h
 - NG_SSCFU_GETDEFPARAM_INFO, 597
 - NG_SSCFU_NODE_TYPE, 597
 - NGM_SSCFU_COOKIE, 597
- ng_sscfu_cmdlist
 - ng_sscfu.c, 616
- ng_sscfu_constructor
 - ng_sscfu.c, 613, 616
- ng_sscfu_cust.h
 - ASSERT, 619
 - DECL_SIGQ_GET, 619
 - MBUF_FREE, 619
 - MEMFREE, 619
 - MEMINIT, 619
 - MEMZALLOC, 619
 - SIG_ALLOC, 620
 - SIG_FREE, 620
 - SIGQ_APPEND, 620
 - SIGQ_CLEAR, 620
 - SIGQ_GET, 620
 - SIGQ_INIT, 620
 - TAILQ_ENTRY, 620
 - TAILQ_HEAD, 620
- ng_sscfu_disconnect
 - ng_sscfu.c, 613, 616
- ng_sscfu_getdefparam, 390
 - mask, 390
 - param, 390
- NG_SSCFU_GETDEFPARAM_INFO
 - ng_sscfu.h, 597
- ng_sscfu_getdefparam_type
 - ng_sscfu.c, 616
- ng_sscfu_getdefparam_type_info
 - ng_sscfu.c, 616
- ng_sscfu_mod_event
 - ng_sscfu.c, 614
- ng_sscfu_newhook
 - ng_sscfu.c, 614, 616
- NG_SSCFU_NODE_TYPE
 - ng_sscfu.h, 597
- ng_sscfu_rcvlower
 - ng_sscfu.c, 614, 616
- ng_sscfu_rcvmsg
 - ng_sscfu.c, 614, 616
- ng_sscfu_rcvupper
 - ng_sscfu.c, 614, 616
- ng_sscfu_shutdown
 - ng_sscfu.c, 614, 616
- ng_sscfu_tpestruct
 - ng_sscfu.c, 617
- ng_sscop.c
 - __FBSIDID, 624
 - DDD, 623
 - flow_lower, 624
 - flow_upper, 624
 - MALLOC_DEFINE, 624
 - MODULE_DEPEND, 624
 - NETGRAPH_INIT, 624
 - ng_sscop_cmdlist, 627
 - ng_sscop_constructor, 624, 627
 - ng_sscop_disconnect, 624, 627
 - ng_sscop_mod_event, 625
 - ng_sscop_newhook, 625, 627
 - ng_sscop_param_type, 627
 - ng_sscop_param_type_info, 628
 - ng_sscop_rcvlower, 625, 628
 - ng_sscop_rcvmanage, 625, 628
 - ng_sscop_rcvmsg, 625, 628
 - ng_sscop_rcvupper, 626, 628
 - ng_sscop_setparam_resp_type, 628
 - ng_sscop_setparam_resp_type_info, 628
 - ng_sscop_setparam_type, 628
 - ng_sscop_setparam_type_info, 629
 - ng_sscop_shutdown, 626, 629
 - ng_sscop_tpestruct, 629
 - sscop_send_lower, 626
 - sscop_send_manage, 626
 - sscop_send_upper, 626
 - sscop_verbose, 627
 - text_status, 627
 - VERBOSE, 623
- ng_sscop.h
 - NGM_SSCOP_DISABLE, 600
 - NGM_SSCOP_ENABLE, 600
 - NGM_SSCOP_GETDEBUG, 600
 - NGM_SSCOP_GETPARAM, 600
 - NGM_SSCOP_GETSTATE, 601
 - NGM_SSCOP_SETDEBUG, 601
 - NGM_SSCOP_SETPARAM, 600

- ng_sscop.h
 - NG_SSCOP_NODE_TYPE, 599
 - NG_SSCOP_PARAM_INFO, 599
 - NG_SSCOP_SETPARAM_INFO, 600
 - NG_SSCOP_SETPARAM_RESP_INFO, 600
 - NGM_SSCOP_COOKIE, 600
- ng_sscop_cmdlist
 - ng_sscop.c, 627
- ng_sscop_constructor
 - ng_sscop.c, 624, 627
- ng_sscop_cust.h
 - ASSERT, 632
 - DECL_MBUF_ALLOC, 632
 - DECL_MSGQ_GET, 632
 - DECL_SIGQ_GET, 633
 - MBUF_ALLOC, 633
 - MBUF_APPEND32, 633
 - MBUF_DUP, 633
 - MBUF_FREE, 633
 - MBUF_GET32, 633
 - MBUF_LEN, 633
 - MBUF_PAD4, 633
 - MBUF_STRIP32, 634
 - MBUF_TRAIL32, 634
 - MBUF_UNPAD, 634
 - MEMFREE, 634
 - MEMINIT, 634
 - MEMZALLOC, 634
 - MSG_ALLOC, 634
 - MSG_FREE, 634
 - MSGQ_APPEND, 634
 - MSGQ_CLEAR, 634
 - MSGQ_EMPTY, 635
 - MSGQ_FOREACH, 635
 - MSGQ_GET, 635
 - MSGQ_INIT, 635
 - MSGQ_INSERT_BEFORE, 635
 - MSGQ_PEEK, 635
 - MSGQ_REMOVE, 635
 - ng_sscop_mbuf_get32, 638
 - ng_sscop_mbuf_pad4, 638
 - ng_sscop_mbuf_strip32, 638
 - ng_sscop_mbuf_trail32, 638
 - SIG_ALLOC, 635
 - SIG_FREE, 635
 - SIGQ_APPEND, 636
 - SIGQ_CLEAR, 636
 - SIGQ_EMPTY, 636
 - SIGQ_GET, 636
 - SIGQ_INIT, 636
 - SIGQ_MOVE, 636
 - SIGQ_PREPEND, 636
 - sscop_timer_t, 638
 - TAILQ_ENTRY, 638
 - TAILQ_HEAD, 638
 - TIMER_FUNC, 637
 - TIMER_INIT, 637
 - TIMER_ISACT, 637
 - TIMER_RESTART, 637
 - TIMER_STOP, 637
- ng_sscop_disconnect
 - ng_sscop.c, 624, 627
- ng_sscop_mbuf_get32
 - ng_sscop_cust.h, 638
- ng_sscop_mbuf_pad4
 - ng_sscop_cust.h, 638
- ng_sscop_mbuf_strip32
 - ng_sscop_cust.h, 638
- ng_sscop_mbuf_trail32
 - ng_sscop_cust.h, 638
- ng_sscop_mod_event
 - ng_sscop.c, 625
- ng_sscop_newhook
 - ng_sscop.c, 625, 627
- NG_SSCOP_NODE_TYPE
 - ng_sscop.h, 599
- NG_SSCOP_PARAM_INFO
 - ng_sscop.h, 599
- ng_sscop_param_type
 - ng_sscfu.c, 617
 - ng_sscop.c, 627
- ng_sscop_param_type_info
 - ng_sscfu.c, 617
 - ng_sscop.c, 628
- ng_sscop_rcvlower
 - ng_sscop.c, 625, 628
- ng_sscop_rcvmanage
 - ng_sscop.c, 625, 628
- ng_sscop_rcvmsg
 - ng_sscop.c, 625, 628
- ng_sscop_rcvupper
 - ng_sscop.c, 626, 628
- ng_sscop_setparam, 391
 - mask, 391
 - param, 391
- NG_SSCOP_SETPARAM_INFO
 - ng_sscop.h, 600
- ng_sscop_setparam_resp, 392
 - error, 392
 - mask, 392
- NG_SSCOP_SETPARAM_RESP_INFO
 - ng_sscop.h, 600
- ng_sscop_setparam_resp_type
 - ng_sscop.c, 628
- ng_sscop_setparam_resp_type_info
 - ng_sscop.c, 628
- ng_sscop_setparam_type
 - ng_sscop.c, 628

- ng_sscop_setparam_type_info
 - ng_sscop.c, 629
- ng_sscop_shutdown
 - ng_sscop.c, 626, 629
- ng_sscop_tpestruct
 - ng_sscop.c, 629
- ng_string_getDefault
 - ng_parse.c, 1464
- ng_string_parse
 - ng_parse.c, 1464
- ng_string_unparse
 - ng_parse.c, 1464
- ng_struct_getAlign
 - ng_parse.c, 1464
- ng_struct_getDefault
 - ng_parse.c, 1465
- ng_struct_parse
 - ng_parse.c, 1465
- ng_struct_unparse
 - ng_parse.c, 1465
- ng_tag.c
 - ERROUT, 1622
 - hinfo_p, 1623
 - M_NETGRAPH_TAG, 1622
 - NETGRAPH_INIT, 1623
 - ng_tag_cmdlist, 1625
 - ng_tag_constructor, 1623, 1625
 - ng_tag_default_in, 1625
 - ng_tag_default_out, 1625
 - ng_tag_disconnect, 1623, 1626
 - ng_tag_hookin_type, 1626
 - ng_tag_hookin_type_fields, 1626
 - ng_tag_hookinary_getLength, 1623
 - ng_tag_hookinary_type, 1626
 - ng_tag_hookout_type, 1626
 - ng_tag_hookout_type_fields, 1626
 - ng_tag_hookoutary_getLength, 1623
 - ng_tag_hookoutary_type, 1627
 - ng_tag_newhook, 1623, 1627
 - ng_tag_rcvdata, 1624, 1627
 - ng_tag_rcvmsg, 1624, 1627
 - ng_tag_setdata_in, 1624
 - ng_tag_setdata_out, 1625
 - ng_tag_shutdown, 1625, 1627
 - tpestruct, 1627
- ng_tag.h
 - NGM_TAG_GET_HOOKIN, 1629
 - NGM_TAG_GET_HOOKOUT, 1629
 - NGM_TAG_SET_HOOKIN, 1629
 - NGM_TAG_SET_HOOKOUT, 1629
- ng_tag.h
 - NG_TAG_HOOKIN_SIZE, 1628
 - NG_TAG_HOOKIN_TYPE_INFO, 1628
 - NG_TAG_HOOKOUT_SIZE, 1628
 - NG_TAG_HOOKOUT_TYPE_INFO, 1629
 - NG_TAG_NODE_TYPE, 1629
 - NGM_TAG_COOKIE, 1629
- ng_tag_cmdlist
 - ng_tag.c, 1625
- ng_tag_constructor
 - ng_tag.c, 1623, 1625
- ng_tag_default_in
 - ng_tag.c, 1625
- ng_tag_default_out
 - ng_tag.c, 1625
- ng_tag_disconnect
 - ng_tag.c, 1623, 1626
- ng_tag_hookin, 393
 - ifMatch, 393
 - ifNotMatch, 393
 - strip, 393
 - tag_cookie, 393
 - tag_data, 393
 - tag_id, 393
 - tag_len, 393
 - thisHook, 394
- NG_TAG_HOOKIN_SIZE
 - ng_tag.h, 1628
- ng_tag_hookin_type
 - ng_tag.c, 1626
- ng_tag_hookin_type_fields
 - ng_tag.c, 1626
- NG_TAG_HOOKIN_TYPE_INFO
 - ng_tag.h, 1628
- ng_tag_hookinary_getLength
 - ng_tag.c, 1623
- ng_tag_hookinary_type
 - ng_tag.c, 1626
- ng_tag_hookinfo, 395
 - hi_match, 395
 - hi_nonmatch, 395
 - in, 395
 - in_tag_cookie, 396
 - in_tag_data, 396
 - in_tag_id, 396
 - in_tag_len, 396
 - out, 396
 - out_tag_cookie, 396
 - out_tag_data, 396
 - out_tag_id, 396
 - out_tag_len, 396
 - strip, 396
- ng_tag_hookout, 397
 - tag_cookie, 397
 - tag_data, 397
 - tag_id, 397
 - tag_len, 397
 - thisHook, 397

- NG_TAG_HOOKOUT_SIZE
 - ng_tag.h, 1628
- ng_tag_hookout_type
 - ng_tag.c, 1626
- ng_tag_hookout_type_fields
 - ng_tag.c, 1626
- NG_TAG_HOOKOUT_TYPE_INFO
 - ng_tag.h, 1629
- ng_tag_hookoutary_getLength
 - ng_tag.c, 1623
- ng_tag_hookoutary_type
 - ng_tag.c, 1627
- ng_tag_newhook
 - ng_tag.c, 1623, 1627
- NG_TAG_NODE_TYPE
 - ng_tag.h, 1629
- NG_TAG_PRIO
 - netgraph.h, 1143
- ng_tag_prio, 398
 - discardability, 398
 - priority, 398
 - tag, 398
- ng_tag_rcvdata
 - ng_tag.c, 1624, 1627
- ng_tag_rcvmsg
 - ng_tag.c, 1624, 1627
- ng_tag_setdata_in
 - ng_tag.c, 1624
- ng_tag_setdata_out
 - ng_tag.c, 1625
- ng_tag_shutdown
 - ng_tag.c, 1625, 1627
- ng_tcpmss.c
 - correct_mss, 1632
 - ERROUT, 1631
 - M_CHECK, 1631
 - NETGRAPH_INIT, 1632
 - ng_tcpmss_cmds, 1633
 - ng_tcpmss_config_type, 1633
 - ng_tcpmss_config_type_fields, 1633
 - ng_tcpmss_constructor, 1632, 1634
 - ng_tcpmss_disconnect, 1632, 1634
 - ng_tcpmss_hookstat_type, 1634
 - ng_tcpmss_hookstat_type_fields, 1634
 - ng_tcpmss_newhook, 1632, 1634
 - ng_tcpmss_rcvdata, 1633, 1634
 - ng_tcpmss_rcvmsg, 1633, 1634
 - ng_tcpmss_tpestruct, 1634
 - TCPMSS_ADJUST_CHECKSUM, 1631
- ng_tcpmss.h
 - NGM_TCPMSS_CLR_STATS, 1637
 - NGM_TCPMSS_CONFIG, 1637
 - NGM_TCPMSS_GET_STATS, 1637
 - NGM_TCPMSS_GETCLR_STATS, 1637
- ng_tcpmss.h
 - NG_TCPMSS_CONFIG_INFO, 1636
 - NG_TCPMSS_HOOKSTAT_INFO, 1636
 - NG_TCPMSS_NODE_TYPE, 1636
 - NGM_TCPMSS_COOKIE, 1637
- ng_tcpmss_cmds
 - ng_tcpmss.c, 1633
- ng_tcpmss_config, 399
 - inHook, 399
 - maxMSS, 399
 - outHook, 399
- NG_TCPMSS_CONFIG_INFO
 - ng_tcpmss.h, 1636
- ng_tcpmss_config_type
 - ng_tcpmss.c, 1633
- ng_tcpmss_config_type_fields
 - ng_tcpmss.c, 1633
- ng_tcpmss_constructor
 - ng_tcpmss.c, 1632, 1634
- ng_tcpmss_disconnect
 - ng_tcpmss.c, 1632, 1634
- ng_tcpmss_hookstat, 400
 - FixedPkts, 400
 - maxMSS, 400
 - Octets, 400
 - Packets, 400
 - SYNPkts, 400
- NG_TCPMSS_HOOKSTAT_INFO
 - ng_tcpmss.h, 1636
- ng_tcpmss_hookstat_type
 - ng_tcpmss.c, 1634
- ng_tcpmss_hookstat_type_fields
 - ng_tcpmss.c, 1634
- ng_tcpmss_newhook
 - ng_tcpmss.c, 1632, 1634
- NG_TCPMSS_NODE_TYPE
 - ng_tcpmss.h, 1636
- ng_tcpmss_rcvdata
 - ng_tcpmss.c, 1633, 1634
- ng_tcpmss_rcvmsg
 - ng_tcpmss.c, 1633, 1634
- ng_tcpmss_tpestruct
 - ng_tcpmss.c, 1634
- ng_tee.c
 - NETGRAPH_INIT, 1639
 - ng_tee_cmds, 1641
 - ng_tee_hookstat_type, 1641
 - ng_tee_hookstat_type_fields, 1641
 - ng_tee_stats_type, 1641
 - ng_tee_stats_type_fields, 1641
 - ng_tee_tpestruct, 1642
 - ngt_close, 1639, 1642
 - ngt_constructor, 1639, 1642
 - ngt_disconnect, 1639, 1642

- ngt_newhook, 1640, 1642
- ngt_rcvdata, 1640, 1642
- ngt_rcvmsg, 1640, 1642
- ngt_shutdown, 1640, 1642
- sc_p, 1639
- ng_tee.h
 - NGM_TEE_CLR_STATS, 1644
 - NGM_TEE_GET_STATS, 1644
 - NGM_TEE_GETCLR_STATS, 1644
- ng_tee.h
 - NG_TEE_HOOK_LEFT, 1643
 - NG_TEE_HOOK_LEFT2RIGHT, 1643
 - NG_TEE_HOOK_RIGHT, 1643
 - NG_TEE_HOOK_RIGHT2LEFT, 1643
 - NG_TEE_HOOKSTAT_INFO, 1644
 - NG_TEE_NODE_TYPE, 1644
 - NG_TEE_STATS_INFO, 1644
 - NGM_TEE_COOKIE, 1644
- ng_tee_cmds
 - ng_tee.c, 1641
- NG_TEE_HOOK_LEFT
 - ng_tee.h, 1643
- NG_TEE_HOOK_LEFT2RIGHT
 - ng_tee.h, 1643
- NG_TEE_HOOK_RIGHT
 - ng_tee.h, 1643
- NG_TEE_HOOK_RIGHT2LEFT
 - ng_tee.h, 1643
- ng_tee_hookstat, 401
 - inFrames, 401
 - inOctets, 401
 - outFrames, 401
 - outOctets, 401
- NG_TEE_HOOKSTAT_INFO
 - ng_tee.h, 1644
- ng_tee_hookstat_type
 - ng_tee.c, 1641
- ng_tee_hookstat_type_fields
 - ng_tee.c, 1641
- NG_TEE_NODE_TYPE
 - ng_tee.h, 1644
- ng_tee_stats, 402
 - left, 402
 - left2right, 402
 - right, 402
 - right2left, 402
- NG_TEE_STATS_INFO
 - ng_tee.h, 1644
- ng_tee_stats_type
 - ng_tee.c, 1641
- ng_tee_stats_type_fields
 - ng_tee.c, 1641
- ng_tee_typestruct
 - ng_tee.c, 1642
- NG_TEXTRESPONSE
 - ng_message.h, 1415
- ng_tty.c
 - ERRROUT, 1647
 - FLG_DEBUG, 1647
 - FLG_DIE, 1647
 - MAX_MBUFQ, 1648
 - NETGRAPH_INIT, 1648
 - ngt_close, 1648
 - ngt_connect, 1649, 1651
 - ngt_constructor, 1649, 1651
 - ngt_disc, 1651
 - ngt_disconnect, 1649, 1652
 - NGT_HIWATER, 1648
 - ngt_input, 1649
 - ngt_ldisc, 1652
 - ngt_mod_event, 1649
 - ngt_newhook, 1649, 1652
 - ngt_open, 1649
 - ngt_rcvdata, 1650, 1652
 - ngt_rcvmsg, 1650, 1652
 - ngt_read, 1650
 - ngt_shutdown, 1650, 1652
 - ngt_start, 1650
 - ngt_timeout, 1651
 - ngt_tioctl, 1651
 - ngt_unit, 1652
 - ngt_write, 1651
 - NGTLOCK, 1648
 - NGTUNLOCK, 1648
 - sc_p, 1648
 - typestruct, 1652
- ng_tty.h
 - NGM_TTY_GET_HOTCHAR, 1654
 - NGM_TTY_SET_HOTCHAR, 1654
- ng_tty.h
 - NG_TTY_DFL_HOTCHAR, 1654
 - NG_TTY_HOOK, 1654
 - NG_TTY_NODE_TYPE, 1654
 - NGM_TTY_COOKIE, 1654
- NG_TTY_DFL_HOTCHAR
 - ng_tty.h, 1654
- NG_TTY_HOOK
 - ng_tty.h, 1654
- NG_TTY_NODE_TYPE
 - ng_tty.h, 1654
- ng_tx_pdu
 - hva_stats_ng, 94
- ng_tx_rawcell
 - hva_stats_ng, 94
- ng_type, 403
 - close, 404
 - cmdlist, 404
 - connect, 404

- constructor, [404](#)
- disconnect, [404](#)
- findhook, [404](#)
- LIST_ENTRY, [404](#)
- mod_event, [404](#)
- name, [404](#)
- newhook, [405](#)
- rcvdata, [405](#)
- rcvmsg, [405](#)
- refs, [405](#)
- shutdown, [405](#)
- version, [405](#)
- ng_typeinfoarray_type_info
 - ng_base.c, [1203](#)
- NG_TYPELEN
 - ng_message.h, [1415](#)
- NG_TYPESIZ
 - ng_message.h, [1415](#)
- ng_ubt.c
 - DRIVER_MODULE, [694](#)
 - MODULE_DEPEND, [694](#)
 - MODULE_VERSION, [694](#)
 - ng_ubt_cmdlist, [702](#)
 - ng_ubt_connect, [694](#), [702](#)
 - ng_ubt_constructor, [694](#), [703](#)
 - ng_ubt_disconnect, [694](#), [703](#)
 - ng_ubt_newhook, [694](#), [703](#)
 - ng_ubt_node_qlen_type, [703](#)
 - ng_ubt_node_qlen_type_fields, [703](#)
 - ng_ubt_node_stat_type, [703](#)
 - ng_ubt_node_stat_type_fields, [703](#)
 - ng_ubt_rcvdata, [694](#), [704](#)
 - ng_ubt_rcvmsg, [695](#), [704](#)
 - ng_ubt_shutdown, [695](#), [704](#)
 - typestruct, [704](#)
 - ubt_bulk_in_complete, [696](#)
 - ubt_bulk_in_complete2, [696](#)
 - ubt_bulk_in_start, [696](#)
 - ubt_bulk_out_complete, [697](#)
 - ubt_bulk_out_complete2, [697](#)
 - ubt_bulk_out_start, [697](#)
 - ubt_intr_complete, [698](#)
 - ubt_intr_complete2, [698](#)
 - ubt_intr_start, [698](#)
 - ubt_isoc_in_complete, [699](#)
 - ubt_isoc_in_complete2, [699](#)
 - ubt_isoc_in_start, [699](#)
 - ubt_isoc_out_complete, [699](#)
 - ubt_isoc_out_complete2, [700](#)
 - ubt_isoc_out_start, [700](#)
 - ubt_modevent, [700](#)
 - ubt_request_complete, [701](#)
 - ubt_request_complete2, [701](#)
 - ubt_request_start, [701](#)
 - ubt_reset, [702](#)
 - USB_ATTACH, [702](#)
 - USB_DECLARE_DRIVER, [702](#)
 - USB_DETACH, [702](#)
 - USB_MATCH, [702](#)
- ng_ubt.h
 - NG_UBT_ALERT_LEVEL, [919](#)
 - NG_UBT_ERR_LEVEL, [919](#)
 - NG_UBT_HOOK, [919](#)
 - NG_UBT_INFO_LEVEL, [920](#)
 - ng_ubt_node_debug_ep, [921](#)
 - ng_ubt_node_dev_nodes_ep, [921](#)
 - NG_UBT_NODE_TYPE, [920](#)
 - NG_UBT_WARN_LEVEL, [920](#)
 - NGM_UBT_COOKIE, [920](#)
 - NGM_UBT_NODE_DEV_NODES, [920](#)
 - NGM_UBT_NODE_GET_DEBUG, [920](#)
 - NGM_UBT_NODE_GET_QLEN, [920](#)
 - NGM_UBT_NODE_GET_STAT, [920](#)
 - NGM_UBT_NODE_QUEUE_ACL, [920](#)
 - NGM_UBT_NODE_QUEUE_CMD, [921](#)
 - NGM_UBT_NODE_QUEUE_SCO, [921](#)
 - NGM_UBT_NODE_RESET_STAT, [921](#)
 - NGM_UBT_NODE_SET_DEBUG, [921](#)
 - NGM_UBT_NODE_SET_QLEN, [921](#)
- NG_UBT_ALERT
 - ng_ubt_var.h, [706](#)
- NG_UBT_ALERT_LEVEL
 - ng_ubt.h, [919](#)
- ng_ubt_cmdlist
 - ng_ubt.c, [702](#)
- ng_ubt_connect
 - ng_ubt.c, [694](#), [702](#)
- ng_ubt_constructor
 - ng_ubt.c, [694](#), [703](#)
- ng_ubt_disconnect
 - ng_ubt.c, [694](#), [703](#)
- NG_UBT_ERR
 - ng_ubt_var.h, [706](#)
- NG_UBT_ERR_LEVEL
 - ng_ubt.h, [919](#)
- NG_UBT_HOOK
 - ng_ubt.h, [919](#)
- NG_UBT_INFO
 - ng_ubt_var.h, [706](#)
- NG_UBT_INFO_LEVEL
 - ng_ubt.h, [920](#)
- NG_UBT_M_PULLUP
 - ng_ubt_var.h, [706](#)
- ng_ubt_newhook
 - ng_ubt.c, [694](#), [703](#)
- ng_ubt_node_debug_ep
 - ng_ubt.h, [921](#)
- ng_ubt_node_dev_nodes_ep

- ng_ubt.h, 921
- ng_ubt_node_qlen_ep, 406
 - qlen, 406
 - queue, 406
- ng_ubt_node_qlen_type
 - ng_ubt.c, 703
- ng_ubt_node_qlen_type_fields
 - ng_ubt.c, 703
- ng_ubt_node_stat_ep, 407
 - bytes_recv, 407
 - bytes_sent, 407
 - ierrors, 407
 - oerrors, 407
 - pckts_recv, 407
 - pckts_sent, 407
- ng_ubt_node_stat_type
 - ng_ubt.c, 703
- ng_ubt_node_stat_type_fields
 - ng_ubt.c, 703
- NG_UBT_NODE_TYPE
 - ng_ubt.h, 920
- ng_ubt_rcvdata
 - ng_ubt.c, 694, 704
- ng_ubt_rcvmsg
 - ng_ubt.c, 695, 704
- ng_ubt_shutdown
 - ng_ubt.c, 695, 704
- NG_UBT_STAT_BYTES_RECV
 - ng_ubt_var.h, 706
- NG_UBT_STAT_BYTES_SENT
 - ng_ubt_var.h, 706
- NG_UBT_STAT_IERROR
 - ng_ubt_var.h, 706
- NG_UBT_STAT_OERROR
 - ng_ubt_var.h, 707
- NG_UBT_STAT_PCKTS_RECV
 - ng_ubt_var.h, 707
- NG_UBT_STAT_PCKTS_SENT
 - ng_ubt_var.h, 707
- NG_UBT_STAT_RESET
 - ng_ubt_var.h, 707
- ng_ubt_var.h
 - NG_UBT_ALERT, 706
 - NG_UBT_ERR, 706
 - NG_UBT_INFO, 706
 - NG_UBT_M_PULLUP, 706
 - NG_UBT_STAT_BYTES_RECV, 706
 - NG_UBT_STAT_BYTES_SENT, 706
 - NG_UBT_STAT_IERROR, 706
 - NG_UBT_STAT_OERROR, 707
 - NG_UBT_STAT_PCKTS_RECV, 707
 - NG_UBT_STAT_PCKTS_SENT, 707
 - NG_UBT_STAT_RESET, 707
 - NG_UBT_WARN, 707
- UBT_ACL_RECV, 707
- UBT_ACL_XMIT, 707
- UBT_ANY_DEV, 707
- UBT_BULK_BUFFER_SIZE, 707
- UBT_BULK_DEV, 708
- UBT_CMD_XMIT, 708
- UBT_CTRL_BUFFER_SIZE, 708
- UBT_CTRL_DEV, 708
- UBT_DEFAULT_QLEN, 708
- UBT_EVT_RECV, 708
- UBT_HAVE_FRAME_TYPE, 708
- UBT_HCI_REQUEST, 708
- UBT_INTR_DEV, 708
- UBT_ISOC_BUFFER_SIZE, 709
- UBT_NEED_FRAME_TYPE, 709
- UBT_SCO_RECV, 709
- UBT_SCO_XMIT, 709
- ubt_softc_p, 709
- ubt_softc_t, 709
- NG_UBT_WARN
 - ng_ubt_var.h, 707
- NG_UBT_WARN_LEVEL
 - ng_ubt.h, 920
- ng_UI.c
 - ERROUT, 1657
 - HDLC_UI, 1657
 - MAX_ENCAPS_HDR, 1657
 - NETGRAPH_INIT, 1657
 - ng_UI_constructor, 1657, 1658
 - ng_UI_disconnect, 1657, 1658
 - ng_UI_newhook, 1658
 - ng_UI_rcvdata, 1658, 1659
 - ng_UI_rcvmsg, 1658, 1659
 - ng_UI_shutdown, 1658, 1659
 - priv_p, 1657
 - typestruct, 1659
- ng_UI.h
 - NG_UI_HOOK_DOWNSTREAM, 1660
 - NG_UI_HOOK_UPSTREAM, 1660
 - NG_UI_NODE_TYPE, 1660
 - NGM_UI_COOKIE, 1660
- ng_UI_constructor
 - ng_UI.c, 1657, 1658
- ng_UI_disconnect
 - ng_UI.c, 1657, 1658
- NG_UI_HOOK_DOWNSTREAM
 - ng_UI.h, 1660
- NG_UI_HOOK_UPSTREAM
 - ng_UI.h, 1660
- ng_UI_newhook
 - ng_UI.c, 1658
- NG_UI_NODE_TYPE
 - ng_UI.h, 1660
- ng_UI_private, 408

- downlink, 408
- uplink, 408
- ng_UI_rcvdata
 - ng_UI.c, 1658, 1659
- ng_UI_rcvmsg
 - ng_UI.c, 1658, 1659
- ng_UI_shutdown
 - ng_UI.c, 1658, 1659
- ng_uncallout
 - netgraph.h, 1157
 - ng_base.c, 1198
- ng_uni.c
 - __FBSDID, 642
 - D, 642
 - dump_saal_signal, 642
 - dump_uni_msg, 642
 - LIST_HEAD, 642
 - MALLOC_DEFINE, 643
 - MODULE_DEPEND, 643
 - NETGRAPH_INIT, 643
 - ng_uni_cmdlist, 646
 - ng_uni_config_mask_type, 646
 - ng_uni_config_mask_type_info, 647
 - ng_uni_config_type, 647
 - ng_uni_config_type_info, 647
 - ng_uni_constructor, 643, 647
 - ng_uni_debug_type, 647
 - ng_uni_debug_type_info, 647
 - ng_uni_debuglevel_type, 648
 - ng_uni_debuglevel_type_info, 648
 - ng_uni_disconnect, 643, 648
 - ng_uni_free, 643
 - ng_uni_malloc, 643
 - ng_uni_mod_event, 643
 - ng_uni_newhook, 644, 648
 - ng_uni_rcvlower, 644, 648
 - ng_uni_rcvmsg, 644, 648
 - ng_uni_rcvupper, 644, 648
 - ng_uni_set_config_type, 648
 - ng_uni_set_config_type_info, 649
 - ng_uni_shutdown, 645, 649
 - ng_uni_typestruct, 649
 - nguni_freemem, 649
 - nguni_unilist_mtx, 649
 - nguni_usedmem, 650
 - text_status, 645
 - UNI_DEBUG_DEFINE, 642
 - uni_do_status, 645
 - uni_fini, 645
 - uni_init, 645
 - uni_saal_output, 645
 - uni_uni_output, 646
 - uni_verbose, 646
 - unimem_names, 650
- ng_uni.h
 - NGM_UNI_DISABLE, 604
 - NGM_UNI_ENABLE, 604
 - NGM_UNI_GET_CONFIG, 604
 - NGM_UNI_GETDEBUG, 604
 - NGM_UNI_GETSTATE, 604
 - NGM_UNI_SET_CONFIG, 604
 - NGM_UNI_SETDEBUG, 604
- ng_uni.h
 - NG_UNI_NODE_TYPE, 602
 - NGM_UNI_CONFIG_INFO, 602
 - NGM_UNI_CONFIG_MASK_INFO, 603
 - NGM_UNI_COOKIE, 603
 - NGM_UNI_DEBUG_INFO, 603
 - NGM_UNI_DEBUGLEVEL_INFO, 603
 - NGM_UNI_SET_CONFIG_INFO, 604
- ng_uni_cmdlist
 - ng_uni.c, 646
- ng_uni_config_mask_type
 - ng_uni.c, 646
- ng_uni_config_mask_type_info
 - ng_uni.c, 647
- ng_uni_config_type
 - ng_uni.c, 647
- ng_uni_config_type_info
 - ng_uni.c, 647
- ng_uni_constructor
 - ng_uni.c, 643, 647
- ng_uni_cust.h
 - UNIMEM_ALL, 656
 - UNIMEM_CALL, 656
 - UNIMEM_INS, 656
 - UNIMEM_PARTY, 656
 - UNIMEM_SIG, 656
- ng_uni_cust.h
 - _TIMER_DESTROY, 652
 - _TIMER_INIT, 652
 - _TIMER_START, 652
 - _TIMER_STOP, 653
 - ASSERT, 653
 - CALL_ALLOC, 653
 - CALL_FREE, 653
 - INS_ALLOC, 653
 - INS_FREE, 653
 - memmove, 653
 - ng_uni_free, 656
 - ng_uni_malloc, 656
 - PARTY_ALLOC, 653
 - PARTY_FREE, 654
 - SIG_ALLOC, 654
 - SIG_FREE, 654
 - TIMER_FUNC_CALL, 654
 - TIMER_FUNC_PARTY, 654
 - TIMER_FUNC_UNI, 654

- TIMER_ISACT, 655
- UNI_ALLOC, 655
- UNI_FREE, 655
- UNICORE, 655
- unimem, 656
- unimem_sizes, 656
- UNIMEM_TYPES, 655
- ng_uni_debug_type
 - ng_uni.c, 647
- ng_uni_debug_type_info
 - ng_uni.c, 647
- ng_uni_debuglevel_type
 - ng_uni.c, 648
- ng_uni_debuglevel_type_info
 - ng_uni.c, 648
- ng_uni_disconnect
 - ng_uni.c, 643, 648
- ng_uni_free
 - ng_uni.c, 643
 - ng_uni_cust.h, 656
- ng_uni_malloc
 - ng_uni.c, 643
 - ng_uni_cust.h, 656
- ng_uni_mod_event
 - ng_uni.c, 643
- ng_uni_newhook
 - ng_uni.c, 644, 648
- NG_UNI_NODE_TYPE
 - ng_uni.h, 602
- ng_uni_rcvlower
 - ng_uni.c, 644, 648
- ng_uni_rcvmsg
 - ng_uni.c, 644, 648
- ng_uni_rcvupper
 - ng_uni.c, 644, 648
- ng_uni_set_config_type
 - ng_uni.c, 648
- ng_uni_set_config_type_info
 - ng_uni.c, 649
- ng_uni_shutdown
 - ng_uni.c, 645, 649
- ng_uni_tpestruct
 - ng_uni.c, 649
- ng_units_in_use
 - ng_fec.c, 1296
 - ng_sppp.c, 1617
- ng_unname
 - ng_base.c, 1198
- ng_unparse
 - ng_parse.c, 1466
 - ng_parse.h, 1480
- ng_unparse_composite
 - ng_parse.c, 1466
- ng_unparse_t
 - ng_parse.h, 1478
- ng_unref_hook
 - netgraph.h, 1157
 - ng_base.c, 1198
- ng_unref_node
 - netgraph.h, 1157
 - ng_base.c, 1198
- ng_vatmpif_config, 409
 - debug, 409
 - macaddr, 409
 - pcr, 409
- ng_vatmpif_hook, 410
 - cur_pcr, 410
 - hook, 410
 - InSeq, 410
 - OutSeq, 411
 - stats, 411
- ng_vatmpif_private
 - ng_atmpif_var.h, 547
- NG_VERSION
 - ng_message.h, 1415
- ng_vjc.c
 - ERROUT, 1663
 - MAX_VJHEADER, 1663
 - NETGRAPH_INIT, 1663
 - ng_vjc_cmds, 1664
 - ng_vjc_config_type, 1664
 - ng_vjc_config_type_fields, 1665
 - ng_vjc_constructor, 1663, 1665
 - ng_vjc_cs_hdr_type, 1665
 - ng_vjc_cs_hdr_type_info, 1665
 - ng_vjc_cstate_type, 1665
 - ng_vjc_cstate_type_fields, 1665
 - ng_vjc_cstatearray_type, 1666
 - ng_vjc_cstatearray_type_info, 1666
 - ng_vjc_disconnect, 1663, 1666
 - ng_vjc_newhook, 1663, 1666
 - ng_vjc_pulluphdrs, 1664
 - ng_vjc_rcvdata, 1664, 1666
 - ng_vjc_rcvmsg, 1664, 1666
 - ng_vjc_shutdown, 1664, 1666
 - ng_vjc_slcompress_type, 1667
 - ng_vjc_slcompress_type_fields, 1667
 - NG_VJC_TSTATE_PTR_TYPE, 1663
 - ng_vjc_tpestruct, 1667
 - priv_p, 1663
- ng_vjc.h
 - NGM_VJC_CLR_STATS, 1670
 - NGM_VJC_GET_CONFIG, 1670
 - NGM_VJC_GET_STATE, 1670
 - NGM_VJC_RECV_ERROR, 1670
 - NGM_VJC_SET_CONFIG, 1670
- ng_vjc.h
 - NG_VJC_CONFIG_TYPE_INFO, 1668

- NG_VJC_HOOK_IP, 1668
- NG_VJC_HOOK_VJCOMP, 1669
- NG_VJC_HOOK_VJIP, 1669
- NG_VJC_HOOK_VJUNCOMP, 1669
- NG_VJC_MAX_CHANNELS, 1669
- NG_VJC_MIN_CHANNELS, 1669
- NG_VJC_NODE_TYPE, 1669
- NGM_VJC_COOKIE, 1669
- ng_vjc_cmds
 - ng_vjc.c, 1664
- ng_vjc_config_type
 - ng_vjc.c, 1664
- ng_vjc_config_type_fields
 - ng_vjc.c, 1665
- NG_VJC_CONFIG_TYPE_INFO
 - ng_vjc.h, 1668
- ng_vjc_constructor
 - ng_vjc.c, 1663, 1665
- ng_vjc_cs_hdr_type
 - ng_vjc.c, 1665
- ng_vjc_cs_hdr_type_info
 - ng_vjc.c, 1665
- ng_vjc_cstate_type
 - ng_vjc.c, 1665
- ng_vjc_cstate_type_fields
 - ng_vjc.c, 1665
- ng_vjc_cstatearray_type
 - ng_vjc.c, 1666
- ng_vjc_cstatearray_type_info
 - ng_vjc.c, 1666
- ng_vjc_disconnect
 - ng_vjc.c, 1663, 1666
- NG_VJC_HOOK_IP
 - ng_vjc.h, 1668
- NG_VJC_HOOK_VJCOMP
 - ng_vjc.h, 1669
- NG_VJC_HOOK_VJIP
 - ng_vjc.h, 1669
- NG_VJC_HOOK_VJUNCOMP
 - ng_vjc.h, 1669
- NG_VJC_MAX_CHANNELS
 - ng_vjc.h, 1669
- NG_VJC_MIN_CHANNELS
 - ng_vjc.h, 1669
- ng_vjc_newhook
 - ng_vjc.c, 1663, 1666
- NG_VJC_NODE_TYPE
 - ng_vjc.h, 1669
- ng_vjc_private, 412
 - conf, 412
 - ip, 412
 - slc, 412
 - vjcomp, 412
 - vjip, 413
 - vjuncomp, 413
- ng_vjc_pulluphdrs
 - ng_vjc.c, 1664
- ng_vjc_rcvdata
 - ng_vjc.c, 1664, 1666
- ng_vjc_rcvmsg
 - ng_vjc.c, 1664, 1666
- ng_vjc_shutdown
 - ng_vjc.c, 1664, 1666
- ng_vjc_slcompress_type
 - ng_vjc.c, 1667
- ng_vjc_slcompress_type_fields
 - ng_vjc.c, 1667
- NG_VJC_TSTATE_PTR_TYPE
 - ng_vjc.c, 1663
- ng_vjc_tpestruct
 - ng_vjc.c, 1667
- ng_vlan.c
 - HASH, 1672
 - HASHSIZE, 1672
 - LIST_HEAD, 1673
 - NETGRAPH_INIT, 1673
 - ng_vlan_cmdlist, 1674
 - ng_vlan_constructor, 1673, 1675
 - ng_vlan_disconnect, 1673, 1675
 - ng_vlan_filter_fields, 1675
 - ng_vlan_filter_type, 1675
 - ng_vlan_findextry, 1673
 - ng_vlan_getTableLength, 1673
 - ng_vlan_newhook, 1673, 1675
 - ng_vlan_rcvdata, 1674, 1675
 - ng_vlan_rcvmsg, 1674, 1676
 - ng_vlan_shutdown, 1674, 1676
 - ng_vlan_table_array_info, 1676
 - ng_vlan_table_array_type, 1676
 - ng_vlan_table_fields, 1676
 - ng_vlan_table_type, 1676
 - ng_vlan_tpestruct, 1676
- ng_vlan.h
 - NGM_VLAN_ADD_FILTER, 1679
 - NGM_VLAN_DEL_FILTER, 1679
 - NGM_VLAN_GET_TABLE, 1679
- ng_vlan.h
 - NG_VLAN_FILTER_FIELDS, 1678
 - NG_VLAN_HOOK_DOWNSTREAM, 1678
 - NG_VLAN_HOOK_NOMATCH, 1678
 - NG_VLAN_NODE_TYPE, 1678
 - NG_VLAN_TABLE_FIELDS, 1679
 - NGM_VLAN_COOKIE, 1679
- ng_vlan_cmdlist
 - ng_vlan.c, 1674
- ng_vlan_constructor
 - ng_vlan.c, 1673, 1675
- ng_vlan_disconnect

- ng_vlan.c, 1673, 1675
- ng_vlan_filter, 414
 - hook, 414
 - vlan, 414
- NG_VLAN_FILTER_FIELDS
 - ng_vlan.h, 1678
- ng_vlan_filter_fields
 - ng_vlan.c, 1675
- ng_vlan_filter_type
 - ng_vlan.c, 1675
- ng_vlan_findex
 - ng_vlan.c, 1673
- ng_vlan_getTableLength
 - ng_vlan.c, 1673
- NG_VLAN_HOOK_DOWNSTREAM
 - ng_vlan.h, 1678
- NG_VLAN_HOOK_NOMATCH
 - ng_vlan.h, 1678
- ng_vlan_newhook
 - ng_vlan.c, 1673, 1675
- NG_VLAN_NODE_TYPE
 - ng_vlan.h, 1678
- ng_vlan_rcvdata
 - ng_vlan.c, 1674, 1675
- ng_vlan_rcvmsg
 - ng_vlan.c, 1674, 1676
- ng_vlan_shutdown
 - ng_vlan.c, 1674, 1676
- ng_vlan_table, 415
 - filter, 415
 - n, 415
- ng_vlan_table_array_info
 - ng_vlan.c, 1676
- ng_vlan_table_array_type
 - ng_vlan.c, 1676
- NG_VLAN_TABLE_FIELDS
 - ng_vlan.h, 1679
- ng_vlan_table_fields
 - ng_vlan.c, 1676
- ng_vlan_table_type
 - ng_vlan.c, 1676
- ng_vlan_tpestruct
 - ng_vlan.c, 1676
- NG_WAITOK
 - netgraph.h, 1143
- ng_worklist_remove
 - ng_base.c, 1198
- NG_WORKQ
 - netgraph.h, 1144
- ng_xxx_cmdlist
 - ng_sample.c, 1576
- ng_xxx_connect
 - ng_sample.c, 1575, 1576
- ng_xxx_constructor
 - ng_sample.c, 1575, 1576
- ng_xxx_disconnect
 - ng_sample.c, 1575, 1576
- NG_XXX_HOOK_DEBUG
 - ng_sample.h, 1578
- NG_XXX_HOOK_DLCI_LEADIN
 - ng_sample.h, 1578
- NG_XXX_HOOK_DOWNSTREAM
 - ng_sample.h, 1578
- ng_xxx_newhook
 - ng_sample.c, 1575, 1576
- NG_XXX_NODE_TYPE
 - ng_sample.h, 1578
- ng_xxx_rcvdata
 - ng_sample.c, 1575, 1577
- ng_xxx_rcvmsg
 - ng_sample.c, 1575, 1577
- ng_xxx_shutdown
 - ng_sample.c, 1576, 1577
- ng_xxx_stat_type
 - ng_sample.c, 1577
- ng_xxx_stat_type_fields
 - ng_sample.c, 1577
- NG_XXX_STATS_TYPE_INFO
 - ng_sample.h, 1578
- nga_async_add
 - ng_async.c, 1160
- nga_cmdlist
 - ng_async.c, 1162
- nga_config_type
 - ng_async.c, 1162
- nga_config_type_fields
 - ng_async.c, 1163
- nga_constructor
 - ng_async.c, 1160, 1163
- nga_disconnect
 - ng_async.c, 1161, 1163
- nga_newhook
 - ng_async.c, 1161, 1163
- nga_rcv_async
 - ng_async.c, 1161
- nga_rcv_sync
 - ng_async.c, 1161
- nga_rcvdata
 - ng_async.c, 1161, 1163
- nga_rcvmsg
 - ng_async.c, 1162, 1163
- nga_shutdown
 - ng_async.c, 1162, 1163
- nga_stats_type
 - ng_async.c, 1163
- nga_stats_type_fields
 - ng_async.c, 1163
- ngatm_data

- ngatmbase.c, 608
- ngatm_handler
 - ngatmbase.c, 607
- ngatm_msg, 416
- ngatm_unilist_mtx
 - ngatmbase.c, 608
- ngatmbase.c
 - __FBSDID, 606
 - DECLARE_MODULE, 606
 - EXTRA, 606
 - LIST_HEAD, 606
 - MALLOC_DEFINE, 606, 607
 - MODULE_VERSION, 607
 - ngatm_data, 608
 - ngatm_handler, 607
 - ngatm_unilist_mtx, 608
 - NGATMBASE_VERSION, 606
 - uni_msg_alloc, 607
 - uni_msg_append, 607
 - uni_msg_build, 607
 - uni_msg_destroy, 607
 - uni_msg_extend, 607
 - uni_msg_init, 608
 - uni_msg_pack_mbuf, 608
 - uni_msg_unpack_mbuf, 608
- ngatmbase.h
 - uni_msg_alloc, 609
 - uni_msg_build, 609
 - uni_msg_destroy, 609
 - uni_msg_pack_mbuf, 609
 - uni_msg_unpack_mbuf, 610
- NGATMBASE_VERSION
 - ngatmbase.c, 606
- ngauto_state_machine
 - ng_lmi.c, 1398
- ngb_mod_event
 - ng_base.c, 1199
- ngc_attach
 - ng_socket.c, 1585
- ngc_bind
 - ng_socket.c, 1586
- ngc_connect
 - ng_socket.c, 1586
- ngc_detach
 - ng_socket.c, 1586
- ngc_send
 - ng_socket.c, 1586
- ngc_usrreqs
 - ng_socket.c, 1590
- ngd_attach
 - ng_socket.c, 1587
- ngd_cdevsw
 - ng_device.c, 1254
- ngd_connect
 - ng_socket.c, 1587
- ngd_detach
 - ng_socket.c, 1587
- ngd_mtx
 - ngd_private, 417
- ngd_private, 417
 - flags, 417
 - hook, 417
 - ngd_mtx, 417
 - ngddev, 418
 - node, 418
 - readq, 418
 - unit, 418
- ngd_send
 - ng_socket.c, 1588
- ngd_typestruct
 - ng_device.c, 1254
- ngd_unit
 - ng_device.c, 1255
- ngd_usrreqs
 - ng_socket.c, 1590
- ngdclose
 - ng_device.c, 1253, 1255
- ngddev
 - ngd_private, 418
- NGDF_OPEN
 - ng_device.c, 1251
- NGDF_RWAIT
 - ng_device.c, 1252
- ngdomain
 - ng_socket.c, 1591
- ngdopen
 - ng_device.c, 1253, 1255
- ngdpoll
 - ng_device.c, 1253, 1255
- ngdread
 - ng_device.c, 1253, 1255
- ngdwrite
 - ng_device.c, 1253, 1255
- nge_cons
 - ng_echo.c, 1258
- nge_disconnect
 - ng_echo.c, 1258
- nge_rcvdata
 - ng_echo.c, 1258
- nge_rcvmsg
 - ng_echo.c, 1258
- NGF_CLOSING
 - netgraph.h, 1144
- NGF_FORCE_WRITER
 - netgraph.h, 1144
- NGF_INVALID
 - netgraph.h, 1144
- NGF_ORIG

- ng_message.h, 1415
- NGF_REALLY_DIE
 - netgraph.h, 1144
- NGF_RESP
 - ng_message.h, 1416
- NGF_TYPE1
 - netgraph.h, 1144
- NGF_TYPE2
 - netgraph.h, 1144
- NGF_TYPE3
 - netgraph.h, 1144
- NGF_TYPE4
 - netgraph.h, 1144
- NGF_WORKQ
 - netgraph.h, 1144
- ngfrm_addrln
 - ng_frame_relay.c, 1303
- ngfrm_allocate_CTX
 - ng_frame_relay.c, 1303
- ngfrm_constructor
 - ng_frame_relay.c, 1303, 1305
- ngfrm_decode
 - ng_frame_relay.c, 1304
- ngfrm_disconnect
 - ng_frame_relay.c, 1304, 1305
- ngfrm_newhook
 - ng_frame_relay.c, 1304, 1305
- ngfrm_rcvdata
 - ng_frame_relay.c, 1304, 1305
- ngfrm_shutdown
 - ng_frame_relay.c, 1305
- ngh_cons
 - ng_hole.c, 1327, 1329
- ngh_disconnect
 - ng_hole.c, 1327, 1329
- ngh_newhook
 - ng_hole.c, 1327, 1329
- ngh_rcvdata
 - ng_hole.c, 1327, 1329
- ngh_rcvmmsg
 - ng_hole.c, 1328, 1329
- NGI_ARG1
 - netgraph.h, 1145
- NGI_ARG2
 - netgraph.h, 1145
- NGI_CLR_HOOK
 - netgraph.h, 1145
- NGI_CLR_NODE
 - netgraph.h, 1145
- NGI_FN
 - netgraph.h, 1145
- NGI_GET_HOOK
 - netgraph.h, 1145
- NGI_GET_M
 - netgraph.h, 1145
- NGI_GET_META
 - netgraph.h, 1146
- NGI_GET_MSG
 - netgraph.h, 1146
- NGI_GET_NODE
 - netgraph.h, 1146
- NGI_HOOK
 - netgraph.h, 1147
- NGI_M
 - netgraph.h, 1147
- NGI_META
 - netgraph.h, 1147
- NGI_MSG
 - netgraph.h, 1147
- NGI_NODE
 - netgraph.h, 1147
- NGI_QUEUED_READER
 - netgraph.h, 1147
- NGI_QUEUED_WRITER
 - netgraph.h, 1147
- NGI_RETADDR
 - netgraph.h, 1147
- NGI_SET_HOOK
 - netgraph.h, 1148
- NGI_SET_NODE
 - netgraph.h, 1148
- NGI_SET_READER
 - netgraph.h, 1148
- NGI_SET_WRITER
 - netgraph.h, 1148
- ngif_name
 - ng_fec_ifname, 202
- ngintr
 - ng_base.c, 1199
- NGIOCGINFO
 - ng_message.h, 1416
- NGIOCSETNAME
 - ng_message.h, 1416
- ngipi_cons
 - ng_ip_input.c, 1348, 1349
- ngipi_disconnect
 - ng_ip_input.c, 1348, 1349
- ngipi_rcvdata
 - ng_ip_input.c, 1349
- nglmi_checkdata
 - ng_lmi.c, 1398
- nglmi_constructor
 - ng_lmi.c, 1399, 1401
- nglmi_disconnect
 - ng_lmi.c, 1399, 1401
- nglmi_inquire
 - ng_lmi.c, 1399
- nglmi_newhook

- ng_lmi.c, 1399, 1401
- nglmi_rcvdata
 - ng_lmi.c, 1400, 1401
- nglmi_rcvmsg
 - ng_lmi.c, 1400, 1401
- nglmi_shutdown
 - ng_lmi.c, 1400, 1402
- nglmi_softc, 419
 - dlci_state, 420
 - flags, 420
 - handle, 420
 - invalidx, 420
 - liv_per_full, 420
 - liv_rate, 420
 - lives, 420
 - lmi_annexA, 420
 - lmi_annexD, 420
 - lmi_channel, 420
 - lmi_channel0, 421
 - lmi_channel1023, 421
 - lmi_group4, 421
 - local_seq, 421
 - need_full, 421
 - node, 421
 - poll_count, 421
 - poll_state, 421
 - protoID, 421
 - protoname, 421
 - remote_seq, 421
 - seq_retries, 422
- nglmi_startup
 - ng_lmi.c, 1400
- nglmi_startup_auto
 - ng_lmi.c, 1401
- nglmi_startup_fixed
 - ng_lmi.c, 1401
- nglmistat, 423
 - autod, 423
 - fixed, 423
 - hook, 423
 - proto, 423
 - seen, 423
 - up, 423
- NGM_ASCII2BINARY
 - ng_message.h, 1418
- NGM_ASYNC_CMD_CLR_STATS
 - ng_async.h, 1167
- NGM_ASYNC_CMD_GET_CONFIG
 - ng_async.h, 1167
- NGM_ASYNC_CMD_GET_STATS
 - ng_async.h, 1167
- NGM_ASYNC_CMD_SET_CONFIG
 - ng_async.h, 1167
- NGM_ASYNC_COOKIE
 - ng_async.h, 1166
- NGM_ATM_ACR_CHANGE
 - ng_atm.h, 587
- ngm_atm_acr_change, 424
 - acr, 424
 - node, 424
 - vci, 424
 - vpi, 424
- NGM_ATM_ACR_CHANGE_INFO
 - ng_atm.h, 583
- NGM_ATM_CARRIER_CHANGE
 - ng_atm.h, 587
- ngm_atm_config, 425
 - max_vccs, 425
 - max_vpcs, 425
 - pcr, 425
 - vci_bits, 425
 - vpi_bits, 425
- NGM_ATM_CONFIG_INFO
 - ng_atm.h, 584
- NGM_ATM_COOKIE
 - ng_atm.h, 584
- NGM_ATM_CPCS_INIT
 - ng_atm.h, 587
- ngm_atm_cpcs_init, 426
 - aal, 426
 - adtf, 426
 - cdf, 426
 - flags, 426
 - icr, 427
 - mbs, 427
 - mcr, 427
 - name, 427
 - nrm, 427
 - pcr, 427
 - rdf, 427
 - rif, 427
 - rmtu, 427
 - scr, 428
 - tbe, 428
 - tmtu, 428
 - traffic, 428
 - trm, 428
 - vci, 428
 - vpi, 428
- NGM_ATM_CPCS_INIT_INFO
 - ng_atm.h, 584
- NGM_ATM_CPCS_TERM
 - ng_atm.h, 587
- ngm_atm_cpcs_term, 429
 - name, 429
- NGM_ATM_CPCS_TERM_INFO
 - ng_atm.h, 585
- NGM_ATM_GET_CONFIG

- ng_atm.h, 587
- NGM_ATM_GET_IFNAME
 - ng_atm.h, 587
- NGM_ATM_GET_STATS
 - ng_atm.h, 587
- NGM_ATM_GET_VCC
 - ng_atm.h, 587
- NGM_ATM_GET_VCCID
 - ng_atm.h, 587
- NGM_ATM_GET_VCCS
 - ng_atm.h, 587
- NGM_ATM_IF_CHANGE
 - ng_atm.h, 587
- ngm_atm_if_change, 430
 - carrier, 430
 - node, 430
 - running, 430
- NGM_ATM_IF_CHANGE_INFO
 - ng_atm.h, 585
- ngm_atm_stats, 431
 - in_errors, 431
 - in_packets, 431
 - out_errors, 431
 - out_packets, 431
- NGM_ATM_STATS_INFO
 - ng_atm.h, 585
- NGM_ATM_TPARAM_INFO
 - ng_atm.h, 585
- NGM_ATM_VCC_CHANGE
 - ng_atm.h, 587
- ngm_atm_vcc_change, 432
 - node, 432
 - state, 432
 - vci, 432
 - vpi, 432
- NGM_ATM_VCC_CHANGE_INFO
 - ng_atm.h, 586
- NGM_ATM_VCC_INFO
 - ng_atm.h, 586
- NGM_ATM_VCCARRAY_INFO
 - ng_atm.h, 586
- NGM_ATM_VCCTABLE_INFO
 - ng_atm.h, 586
- NGM_ATMLLC_COOKIE
 - ng_atmllc.h, 1174
- NGM_ATMPIF_CLR_STATS
 - ng_atmpif.h, 592
- NGM_ATMPIF_COOKIE
 - ng_atmpif.h, 590
- NGM_ATMPIF_GET_CONFIG
 - ng_atmpif.h, 592
- NGM_ATMPIF_GET_LINK_STATUS
 - ng_atmpif.h, 592
- NGM_ATMPIF_GET_STATS
 - ng_atmpif.h, 592
- NGM_ATMPIF_GETCLR_STATS
 - ng_atmpif.h, 592
- NGM_ATMPIF_SET_CONFIG
 - ng_atmpif.h, 592
- ngm_bandwidth, 433
 - nominal_in, 433
 - nominal_out, 433
 - seen_in, 433
 - seen_out, 433
- NGM_BINARY2ASCII
 - ng_message.h, 1418
- NGM_BPF_CLR_STATS
 - ng_bpf.h, 1212
- NGM_BPF_COOKIE
 - ng_bpf.h, 1212
- NGM_BPF_GET_PROGRAM
 - ng_bpf.h, 1212
- NGM_BPF_GET_STATS
 - ng_bpf.h, 1212
- NGM_BPF_GETCLR_STATS
 - ng_bpf.h, 1212
- NGM_BPF_SET_PROGRAM
 - ng_bpf.h, 1212
- NGM_BRIDGE_CLR_STATS
 - ng_bridge.h, 1227
- NGM_BRIDGE_COOKIE
 - ng_bridge.h, 1227
- NGM_BRIDGE_GET_CONFIG
 - ng_bridge.h, 1227
- NGM_BRIDGE_GET_STATS
 - ng_bridge.h, 1227
- NGM_BRIDGE_GET_TABLE
 - ng_bridge.h, 1227
- NGM_BRIDGE_GETCLR_STATS
 - ng_bridge.h, 1227
- NGM_BRIDGE_RESET
 - ng_bridge.h, 1227
- NGM_BRIDGE_SET_CONFIG
 - ng_bridge.h, 1227
- NGM_BT3C_COOKIE
 - ng_bt3c.h, 792
- NGM_BT3C_NODE_DOWNLOAD_FIRMWARE
 - ng_bt3c.h, 792
- NGM_BT3C_NODE_GET_DEBUG
 - ng_bt3c.h, 792
- NGM_BT3C_NODE_GET_QLEN
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_GET_STAT
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_GET_STATE
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_IN_QUEUE
 - ng_bt3c.h, 793

- NGM_BT3C_NODE_OUT_QUEUE
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_RESET_STAT
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_SET_DEBUG
 - ng_bt3c.h, 793
- NGM_BT3C_NODE_SET_QLEN
 - ng_bt3c.h, 793
- NGM_CCATM_ADDR_ARRAY_INFO
 - ng_ccatm.h, 593
- ngm_ccatm_addr_req, 434
 - addr, 434
 - port, 434
- NGM_CCATM_ADDR_REQ_ARRAY_INFO
 - ng_ccatm.h, 594
- NGM_CCATM_ADDR_REQ_INFO
 - ng_ccatm.h, 594
- NGM_CCATM_ADDRESS_REGISTERED
 - ng_ccatm.h, 596
- NGM_CCATM_ADDRESS_UNREGISTERED
 - ng_ccatm.h, 596
- NGM_CCATM_ATM_PORT_INFO
 - ng_ccatm.h, 594
- NGM_CCATM_CLEAR
 - ng_ccatm.h, 596
- NGM_CCATM_COOKIE
 - ng_ccatm.h, 594
- NGM_CCATM_DUMP
 - ng_ccatm.h, 596
- NGM_CCATM_ESI_INFO
 - ng_ccatm.h, 595
- NGM_CCATM_GET_ADDRESSES
 - ng_ccatm.h, 596
- ngm_ccatm_get_addresses, 435
 - addr, 435
 - count, 435
- NGM_CCATM_GET_ADDRESSES_INFO
 - ng_ccatm.h, 595
- NGM_CCATM_GET_EXSTAT
 - ng_ccatm.h, 596
- NGM_CCATM_GET_PORT_PARAM
 - ng_ccatm.h, 596
- NGM_CCATM_GET_PORTLIST
 - ng_ccatm.h, 596
- NGM_CCATM_GETSTATE
 - ng_ccatm.h, 596
- ngm_ccatm_port, 436
 - port, 436
- NGM_CCATM_PORT_ARRAY_INFO
 - ng_ccatm.h, 595
- NGM_CCATM_PORT_INFO
 - ng_ccatm.h, 595
- ngm_ccatm_portlist, 437
 - nports, 437
 - ports, 437
- NGM_CCATM_PORTLIST_INFO
 - ng_ccatm.h, 595
- NGM_CCATM_RESET
 - ng_ccatm.h, 596
- NGM_CCATM_SET_PORT_PARAM
 - ng_ccatm.h, 596
- NGM_CCATM_SETLOG
 - ng_ccatm.h, 596
- NGM_CCATM_START
 - ng_ccatm.h, 596
- NGM_CCATM_STOP
 - ng_ccatm.h, 596
- NGM_CCATM_UNI_ADDR_INFO
 - ng_ccatm.h, 596
- NGM_CISCO_COOKIE
 - ng_cisco.h, 1238
- NGM_CISCO_GET_IPADDR
 - ng_cisco.h, 1239
- NGM_CISCO_GET_STATUS
 - ng_cisco.h, 1239
- NGM_CISCO_SET_IPADDR
 - ng_cisco.h, 1239
- NGM_CONNECT
 - ng_message.h, 1418
- ngm_connect, 438
 - ourhook, 438
 - path, 438
 - peerhook, 438
- NGM_DEFLATE_CLR_STATS
 - ng_deflate.h, 1248
- NGM_DEFLATE_CONFIG
 - ng_deflate.h, 1248
- NGM_DEFLATE_COOKIE
 - ng_deflate.h, 1248
- NGM_DEFLATE_GET_STATS
 - ng_deflate.h, 1248
- NGM_DEFLATE_GETCLR_STATS
 - ng_deflate.h, 1248
- NGM_DEFLATE_RESETRQ
 - ng_deflate.h, 1248
- NGM_DEVICE_COOKIE
 - ng_device.h, 1256
- NGM_DEVICE_GET_DEVNAME
 - ng_device.h, 1256
- NGM_DROP_LINK
 - ng_message.h, 1416
- NGM_ECHO_COOKIE
 - ng_echo.h, 1260
- NGM_EIFACE_COOKIE
 - ng_eiface.h, 1269
- NGM_EIFACE_GET_IFADDRS
 - ng_eiface.h, 1269
- NGM_EIFACE_GET_IFNAME

- ng_eiface.h, 1269
- NGM_EIFACE_SET
 - ng_eiface.h, 1269
- NGM ETF_COOKIE
 - ng_etf.h, 1277
- NGM ETF_GET_STATUS
 - ng_etf.h, 1277
- NGM ETF_SET_FILTER
 - ng_etf.h, 1277
- NGM ETF_SET_FLAG
 - ng_etf.h, 1277
- NGM_ETHER_ADD_MULTI
 - ng_ether.h, 1287
- NGM_ETHER_COOKIE
 - ng_ether.h, 1287
- NGM_ETHER_DEL_MULTI
 - ng_ether.h, 1287
- NGM_ETHER_DETACH
 - ng_ether.h, 1287
- NGM_ETHER_GET_AUTOSRC
 - ng_ether.h, 1287
- NGM_ETHER_GET_ENADDR
 - ng_ether.h, 1287
- NGM_ETHER_GET_IFINDEX
 - ng_ether.h, 1287
- NGM_ETHER_GET_IFNAME
 - ng_ether.h, 1287
- NGM_ETHER_GET_PROMISC
 - ng_ether.h, 1287
- NGM_ETHER_SET_AUTOSRC
 - ng_ether.h, 1287
- NGM_ETHER_SET_ENADDR
 - ng_ether.h, 1287
- NGM_ETHER_SET_PROMISC
 - ng_ether.h, 1287
- NGM_FEC_ADD_IFACE
 - ng_fec.h, 1299
- NGM_FEC_COOKIE
 - ng_fec.h, 1298
- NGM_FEC_DEL_IFACE
 - ng_fec.h, 1299
- NGM_FEC_SET_MODE_INET
 - ng_fec.h, 1299
- NGM_FEC_SET_MODE_INET6
 - ng_fec.h, 1299
- NGM_FEC_SET_MODE_MAC
 - ng_fec.h, 1299
- NGM_FLOW_COOKIE
 - ng_message.h, 1416
- NGM_FLUSH_QUEUE
 - ng_message.h, 1416
- NGM_FRAMERELAY_COOKIE
 - ng_frame_relay.h, 1307
- NGM_GENERIC_COOKIE
 - ng_message.h, 1416
- NGM_GET_BANDWIDTH
 - ng_message.h, 1416
- NGM_GET_XMIT_Q_LIMITS
 - ng_message.h, 1417
- NGM_GIF_COOKIE
 - ng_gif.h, 1316
- NGM_GIF_DEMUX_COOKIE
 - ng_gif_demux.h, 1325
- NGM_GIF_GET_IFINDEX
 - ng_gif.h, 1317
- NGM_GIF_GET_IFNAME
 - ng_gif.h, 1317
- NGM_H4_COOKIE
 - ng_h4.h, 844
- NGM_H4_NODE_GET_DEBUG
 - ng_h4.h, 844
- NGM_H4_NODE_GET_QLEN
 - ng_h4.h, 844
- NGM_H4_NODE_GET_STAT
 - ng_h4.h, 845
- NGM_H4_NODE_GET_STATE
 - ng_h4.h, 845
- NGM_H4_NODE_RESET
 - ng_h4.h, 845
- NGM_H4_NODE_RESET_STAT
 - ng_h4.h, 845
- NGM_H4_NODE_SET_DEBUG
 - ng_h4.h, 845
- NGM_H4_NODE_SET_QLEN
 - ng_h4.h, 845
- NGM_HASREPLY
 - ng_message.h, 1417
- NGM_HCI_COOKIE
 - ng_hci.h, 893
- NGM_HCI_LP_CON_CFM
 - ng_hci.h, 893
- NGM_HCI_LP_CON_IND
 - ng_hci.h, 893
- NGM_HCI_LP_CON_REQ
 - ng_hci.h, 893
- NGM_HCI_LP_CON_RSP
 - ng_hci.h, 893
- NGM_HCI_LP_DISCON_IND
 - ng_hci.h, 893
- NGM_HCI_LP_DISCON_REQ
 - ng_hci.h, 893
- NGM_HCI_LP_QOS_CFM
 - ng_hci.h, 894
- NGM_HCI_LP_QOS_IND
 - ng_hci.h, 894
- NGM_HCI_LP_QOS_REQ
 - ng_hci.h, 894

- NGM_HCI_NODE_FLUSH_NEIGHBOR_CACHE
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_BDADDR
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_BUFFER
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_CON_LIST
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_DEBUG
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_FEATURES
 - ng_hci.h, 894
- NGM_HCI_NODE_GET_LINK_POLICY_SETTINGS_MASK
 - ng_hci.h, 895
- NGM_HCI_NODE_GET_NEIGHBOR_CACHE
 - ng_hci.h, 895
- NGM_HCI_NODE_GET_PACKET_MASK
 - ng_hci.h, 895
- NGM_HCI_NODE_GET_ROLE_SWITCH
 - ng_hci.h, 895
- NGM_HCI_NODE_GET_STAT
 - ng_hci.h, 895
- NGM_HCI_NODE_GET_STATE
 - ng_hci.h, 895
- NGM_HCI_NODE_INIT
 - ng_hci.h, 895
- NGM_HCI_NODE_LIST_NAMES
 - ng_hci.h, 895
- NGM_HCI_NODE_RESET_STAT
 - ng_hci.h, 895
- NGM_HCI_NODE_SET_DEBUG
 - ng_hci.h, 896
- NGM_HCI_NODE_SET_LINK_POLICY_SETTINGS_MASK
 - ng_hci.h, 896
- NGM_HCI_NODE_SET_PACKET_MASK
 - ng_hci.h, 896
- NGM_HCI_NODE_SET_ROLE_SWITCH
 - ng_hci.h, 896
- NGM_HCI_NODE_UP
 - ng_hci.h, 896
- NGM_HCI_SYNC_CON_QUEUE
 - ng_hci.h, 896
- NGM_HIGH_WATER_PASSED
 - ng_message.h, 1417
- NGM_HOLE_CLR_STATS
 - ng_hole.h, 1331
- NGM_HOLE_COOKIE
 - ng_hole.h, 1330
- NGM_HOLE_GET_STATS
 - ng_hole.h, 1331
- NGM_HOLE_GETCLR_STATS
 - ng_hole.h, 1331
- ng_hole.h, 1331
- NGM_HUB_COOKIE
 - ng_hub.h, 1335
- NGM_IFACE_BROADCAST
 - ng_iface.h, 1346
- NGM_IFACE_COOKIE
 - ng_iface.h, 1346
- NGM_IFACE_GET_IFINDEX
 - ng_iface.h, 1346
- NGM_IFACE_GET_IFNAME
 - ng_iface.h, 1346
- NGM_IFACE_POINT2POINT
 - ng_iface.h, 1346
- NGM_IP_INPUT_COOKIE
 - ng_ip_input.h, 1350
- NGM_IPFW_COOKIE
 - ng_ipfw.h, 1357
- NGM_KSOCKET_ACCEPT
 - ng_ksocket.h, 1373
- NGM_KSOCKET_ACCEPT_INFO
 - ng_ksocket.h, 1373
- NGM_KSOCKET_BIND
 - ng_ksocket.h, 1373
- NGM_KSOCKET_CONNECT
 - ng_ksocket.h, 1373
- NGM_KSOCKET_COOKIE
 - ng_ksocket.h, 1373
- NGM_KSOCKET_GETNAME
 - ng_ksocket.h, 1373
- NGM_KSOCKET_GETOPT
 - ng_ksocket.h, 1373
- NGM_KSOCKET_GETPEERNAME
 - ng_ksocket.h, 1373
- NGM_KSOCKET_LISTEN
 - ng_ksocket.h, 1373
- NGM_KSOCKET_SETOPT
 - ng_ksocket.h, 1373
- NGM_L2CAP_COOKIE
 - ng_l2cap.h, 914
- NGM_L2CAP_L2CA_CFG
 - ng_l2cap.h, 914
- NGM_L2CAP_L2CA_CFG_IND
 - ng_l2cap.h, 914
- NGM_L2CAP_L2CA_CFG_RSP
 - ng_l2cap.h, 914
- NGM_L2CAP_L2CA_CON
 - ng_l2cap.h, 914
- NGM_L2CAP_L2CA_CON_IND
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_CON_RSP
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_DISCON
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_DISCON_IND

- ng_l2cap.h, 915
- NGM_L2CAP_L2CA_ENABLE_CLT
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_GET_INFO
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_GRP_ADD_MEMBER
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_GRP_CLOSE
 - ng_l2cap.h, 915
- NGM_L2CAP_L2CA_GRP_CREATE
 - ng_l2cap.h, 916
- NGM_L2CAP_L2CA_GRP_MEMBERSHIP
 - ng_l2cap.h, 916
- NGM_L2CAP_L2CA_GRP_REM_MEMBER
 - ng_l2cap.h, 916
- NGM_L2CAP_L2CA_PING
 - ng_l2cap.h, 916
- NGM_L2CAP_L2CA_QOS_IND
 - ng_l2cap.h, 916
- NGM_L2CAP_L2CA_WRITE
 - ng_l2cap.h, 916
- NGM_L2CAP_NODE_GET_AUTO_DISCON_-
 - TIMO
 - ng_l2cap.h, 916
- NGM_L2CAP_NODE_GET_CHAN_LIST
 - ng_l2cap.h, 916
- NGM_L2CAP_NODE_GET_CON_LIST
 - ng_l2cap.h, 917
- NGM_L2CAP_NODE_GET_DEBUG
 - ng_l2cap.h, 917
- NGM_L2CAP_NODE_GET_FLAGS
 - ng_l2cap.h, 917
- NGM_L2CAP_NODE_HOOK_INFO
 - ng_l2cap.h, 917
- NGM_L2CAP_NODE_SET_AUTO_DISCON_-
 - TIMO
 - ng_l2cap.h, 917
- NGM_L2CAP_NODE_SET_DEBUG
 - ng_l2cap.h, 917
- NGM_L2TP_ACK_FAILURE
 - ng_l2tp.h, 1392
- NGM_L2TP_CLR_SESSION_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_CLR_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_COOKIE
 - ng_l2tp.h, 1391
- NGM_L2TP_GET_CONFIG
 - ng_l2tp.h, 1392
- NGM_L2TP_GET_SESS_CONFIG
 - ng_l2tp.h, 1392
- NGM_L2TP_GET_SESSION_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_GET_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_GETCLR_SESSION_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_GETCLR_STATS
 - ng_l2tp.h, 1392
- NGM_L2TP_SET_CONFIG
 - ng_l2tp.h, 1392
- NGM_L2TP_SET_SEQ
 - ng_l2tp.h, 1392
- NGM_L2TP_SET_SESS_CONFIG
 - ng_l2tp.h, 1392
- NGM_LINK_IS_DOWN
 - ng_message.h, 1417
- NGM_LINK_IS_UP
 - ng_message.h, 1417
- NGM_LISTHOOKS
 - ng_message.h, 1418
- NGM_LISTNAMES
 - ng_message.h, 1418
- NGM_LISTNODES
 - ng_message.h, 1418
- NGM_LISTTYPES
 - ng_message.h, 1418
- NGM_LMI_COOKIE
 - ng_lmi.h, 1404
- NGM_LMI_GET_STATUS
 - ng_lmi.h, 1405
- NGM_LMI_STAT_ARYSIZE
 - ng_lmi.h, 1404
- NGM_LOW_WATER_PASSED
 - ng_message.h, 1417
- NGM_MICROMANAGE
 - ng_message.h, 1417
- NGM_MKPEER
 - ng_message.h, 1418
- ngm_mkpeer, 439
 - ourhook, 439
 - peerhook, 439
 - type, 439
- NGM_MPPC_CONFIG_COMP
 - ng_mppc.h, 1429
- NGM_MPPC_CONFIG_DECOMP
 - ng_mppc.h, 1429
- NGM_MPPC_COOKIE
 - ng_mppc.h, 1429
- NGM_MPPC_RESETRREQ
 - ng_mppc.h, 1429
- NGM_NAME
 - ng_message.h, 1418
- ngm_name, 440
 - name, 440
- NGM_NAT_COOKIE
 - ng_nat.h, 1435
- NGM_NAT_SET_IPADDR

- ng_nat.h, 1435
- NGM_NETFLOW_COOKIE
 - ng_netflow.h, 1112
- NGM_NETFLOW_IFINFO
 - ng_netflow.h, 1113
- NGM_NETFLOW_INFO
 - ng_netflow.h, 1113
- NGM_NETFLOW_SETDLT
 - ng_netflow.h, 1113
- NGM_NETFLOW_SETIFINDEX
 - ng_netflow.h, 1113
- NGM_NETFLOW_SETTIMEOUTS
 - ng_netflow.h, 1113
- NGM_NETFLOW_SHOW
 - ng_netflow.h, 1113
- NGM_NODEINFO
 - ng_message.h, 1418
- NGM_ONE2MANY_CLR_STATS
 - ng_one2many.h, 1445
- NGM_ONE2MANY_COOKIE
 - ng_one2many.h, 1445
- NGM_ONE2MANY_GET_CONFIG
 - ng_one2many.h, 1445
- NGM_ONE2MANY_GET_STATS
 - ng_one2many.h, 1445
- NGM_ONE2MANY_GETCLR_STATS
 - ng_one2many.h, 1445
- NGM_ONE2MANY_SET_CONFIG
 - ng_one2many.h, 1445
- NGM_PPP_CLR_LINK_STATS
 - ng_ppp.h, 1516
- NGM_PPP_COOKIE
 - ng_ppp.h, 1515
- NGM_PPP_GET_CONFIG
 - ng_ppp.h, 1516
- NGM_PPP_GET_LINK_STATS
 - ng_ppp.h, 1516
- NGM_PPP_GET_MP_STATE
 - ng_ppp.h, 1516
- NGM_PPP_GETCLR_LINK_STATS
 - ng_ppp.h, 1516
- NGM_PPP_SET_CONFIG
 - ng_ppp.h, 1516
- NGM_PPPOE_ACNAME
 - ng_pppoe.h, 1537
- NGM_PPPOE_CLOSE
 - ng_pppoe.h, 1537
- NGM_PPPOE_CONNECT
 - ng_pppoe.h, 1537
- NGM_PPPOE_COOKIE
 - ng_pppoe.h, 1533
- NGM_PPPOE_FAIL
 - ng_pppoe.h, 1537
- NGM_PPPOE_GET_STATUS
 - ng_pppoe.h, 1537
- NGM_PPPOE_GETMODE
 - ng_pppoe.h, 1537
- NGM_PPPOE_LISTEN
 - ng_pppoe.h, 1537
- NGM_PPPOE_OFFER
 - ng_pppoe.h, 1537
- NGM_PPPOE_SERVICE
 - ng_pppoe.h, 1537
- NGM_PPPOE_SESSIONID
 - ng_pppoe.h, 1537
- NGM_PPPOE_SET_FLAG
 - ng_pppoe.h, 1537
- NGM_PPPOE_SETENADDR
 - ng_pppoe.h, 1537
- NGM_PPPOE_SETMODE
 - ng_pppoe.h, 1537
- NGM_PPPOE_SUCCESS
 - ng_pppoe.h, 1537
- NGM_PPTPGRE_CLR_STATS
 - ng_pptpgre.h, 1553
- NGM_PPTPGRE_COOKIE
 - ng_pptpgre.h, 1552
- NGM_PPTPGRE_GET_CONFIG
 - ng_pptpgre.h, 1552
- NGM_PPTPGRE_GET_STATS
 - ng_pptpgre.h, 1553
- NGM_PPTPGRE_GETCLR_STATS
 - ng_pptpgre.h, 1553
- NGM_PPTPGRE_SET_CONFIG
 - ng_pptpgre.h, 1552
- NGM_PRED1_CLR_STATS
 - ng_pred1.h, 1563
- NGM_PRED1_CONFIG
 - ng_pred1.h, 1563
- NGM_PRED1_COOKIE
 - ng_pred1.h, 1563
- NGM_PRED1_GET_STATS
 - ng_pred1.h, 1563
- NGM_PRED1_GETCLR_STATS
 - ng_pred1.h, 1563
- NGM_PRED1_RESETRQ
 - ng_pred1.h, 1563
- ngm_queue_state, 441
 - current, 441
 - high_watermark, 441
 - low_watermark, 441
 - max_queuelen_bytes, 441
 - max_queuelen_packets, 441
 - queue_priority, 441
- NGM_RAISE_LINK
 - ng_message.h, 1417
- NGM_READONLY
 - ng_message.h, 1417

- NGM_RFC1490_COOKIE
 ng_rfc1490.h, 1571
 NGM_RFC1490_GET_ENCAP
 ng_rfc1490.h, 1572
 NGM_RFC1490_SET_ENCAP
 ng_rfc1490.h, 1572
 NGM_RMHOOK
 ng_message.h, 1418
 ngm_rmhook, 443
 ourhook, 443
 NGM_SET_FLOW_MANAGER
 ng_message.h, 1418
 NGM_SET_XMIT_Q_LIMITS
 ng_message.h, 1418
 NGM_SHUTDOWN
 ng_message.h, 1418
 NGM_SOCKET_CMD_LINGER
 ng_socket.h, 1594
 NGM_SOCKET_CMD_NOLINGER
 ng_socket.h, 1594
 NGM_SOCKET_COOKIE
 ng_socket.h, 1593
 NGM_SOURCE_CLR_DATA
 ng_source.h, 1606
 NGM_SOURCE_CLR_STATS
 ng_source.h, 1606
 NGM_SOURCE_COOKIE
 ng_source.h, 1606
 NGM_SOURCE_GET_STATS
 ng_source.h, 1606
 NGM_SOURCE_GETCLR_STATS
 ng_source.h, 1606
 NGM_SOURCE_SETIFACE
 ng_source.h, 1606
 NGM_SOURCE_SETPPS
 ng_source.h, 1606
 NGM_SOURCE_START
 ng_source.h, 1606
 NGM_SOURCE_STOP
 ng_source.h, 1606
 NGM_SPLIT_COOKIE
 ng_split.h, 1611
 NGM_SPPP_COOKIE
 ng_sppp.h, 1619
 NGM_SPPP_GET_IFNAME
 ng_sppp.h, 1619
 NGM_SSCFU_COOKIE
 ng_sscfu.h, 598
 NGM_SSCFU_DISABLE
 ng_sscfu.h, 598
 NGM_SSCFU_ENABLE
 ng_sscfu.h, 598
 NGM_SSCFU_GETDEBUG
 ng_sscfu.h, 598
 NGM_SSCFU_GETDEFPARAM
 ng_sscfu.h, 598
 NGM_SSCFU_GETSTATE
 ng_sscfu.h, 598
 NGM_SSCFU_SETDEBUG
 ng_sscfu.h, 598
 NGM_SSCOP_COOKIE
 ng_sscop.h, 600
 NGM_SSCOP_DISABLE
 ng_sscop.h, 600
 NGM_SSCOP_ENABLE
 ng_sscop.h, 600
 NGM_SSCOP_GETDEBUG
 ng_sscop.h, 600
 NGM_SSCOP_GETPARAM
 ng_sscop.h, 600
 NGM_SSCOP_GETSTATE
 ng_sscop.h, 601
 NGM_SSCOP_SETDEBUG
 ng_sscop.h, 601
 NGM_SSCOP_SETPARAM
 ng_sscop.h, 600
 NGM_SYNC_QUEUE_STATE
 ng_message.h, 1418
 NGM_TAG_COOKIE
 ng_tag.h, 1629
 NGM_TAG_GET_HOOKIN
 ng_tag.h, 1629
 NGM_TAG_GET_HOOKOUT
 ng_tag.h, 1629
 NGM_TAG_SET_HOOKIN
 ng_tag.h, 1629
 NGM_TAG_SET_HOOKOUT
 ng_tag.h, 1629
 NGM_TCPMSS_CLR_STATS
 ng_tcpmss.h, 1637
 NGM_TCPMSS_CONFIG
 ng_tcpmss.h, 1637
 NGM_TCPMSS_COOKIE
 ng_tcpmss.h, 1637
 NGM_TCPMSS_GET_STATS
 ng_tcpmss.h, 1637
 NGM_TCPMSS_GETCLR_STATS
 ng_tcpmss.h, 1637
 NGM_TEE_CLR_STATS
 ng_tee.h, 1644
 NGM_TEE_COOKIE
 ng_tee.h, 1644
 NGM_TEE_GET_STATS
 ng_tee.h, 1644
 NGM_TEE_GETCLR_STATS
 ng_tee.h, 1644
 NGM_TEXT_CONFIG
 ng_message.h, 1418

- NGM_TEXT_STATUS
 - ng_message.h, 1418
- NGM_TTY_COOKIE
 - ng_tty.h, 1654
- NGM_TTY_GET_HOTCHAR
 - ng_tty.h, 1654
- NGM_TTY_SET_HOTCHAR
 - ng_tty.h, 1654
- NGM_UBT_COOKIE
 - ng_ubt.h, 920
- NGM_UBT_NODE_DEV_NODES
 - ng_ubt.h, 920
- NGM_UBT_NODE_GET_DEBUG
 - ng_ubt.h, 920
- NGM_UBT_NODE_GET_QLEN
 - ng_ubt.h, 920
- NGM_UBT_NODE_GET_STAT
 - ng_ubt.h, 920
- NGM_UBT_NODE_QUEUE_ACL
 - ng_ubt.h, 920
- NGM_UBT_NODE_QUEUE_CMD
 - ng_ubt.h, 921
- NGM_UBT_NODE_QUEUE_SCO
 - ng_ubt.h, 921
- NGM_UBT_NODE_RESET_STAT
 - ng_ubt.h, 921
- NGM_UBT_NODE_SET_DEBUG
 - ng_ubt.h, 921
- NGM_UBT_NODE_SET_QLEN
 - ng_ubt.h, 921
- NGM_UI_COOKIE
 - ng_ui.h, 1660
- NGM_UNI_CONFIG_INFO
 - ng_uni.h, 602
- ngm_uni_config_mask, 444
 - mask, 444
 - option_mask, 444
 - popt_mask, 444
- NGM_UNI_CONFIG_MASK_INFO
 - ng_uni.h, 603
- NGM_UNI_COOKIE
 - ng_uni.h, 603
- ngm_uni_debug, 445
 - level, 445
- NGM_UNI_DEBUG_INFO
 - ng_uni.h, 603
- NGM_UNI_DEBUGLEVEL_INFO
 - ng_uni.h, 603
- NGM_UNI_DISABLE
 - ng_uni.h, 604
- NGM_UNI_ENABLE
 - ng_uni.h, 604
- NGM_UNI_GET_CONFIG
 - ng_uni.h, 604
- NGM_UNI_GETDEBUG
 - ng_uni.h, 604
- NGM_UNI_GETSTATE
 - ng_uni.h, 604
- NGM_UNI_SET_CONFIG
 - ng_uni.h, 604
- ngm_uni_set_config, 446
 - config, 446
 - mask, 446
- NGM_UNI_SET_CONFIG_INFO
 - ng_uni.h, 604
- NGM_UNI_SETDEBUG
 - ng_uni.h, 604
- NGM_VJC_CLR_STATS
 - ng_vjc.h, 1670
- ngm_vjc_config, 447
 - compressCID, 447
 - enableComp, 447
 - enableDecomp, 447
 - maxChannel, 447
- NGM_VJC_COOKIE
 - ng_vjc.h, 1669
- NGM_VJC_GET_CONFIG
 - ng_vjc.h, 1670
- NGM_VJC_GET_STATE
 - ng_vjc.h, 1670
- NGM_VJC_RECV_ERROR
 - ng_vjc.h, 1670
- NGM_VJC_SET_CONFIG
 - ng_vjc.h, 1670
- NGM_VLAN_ADD_FILTER
 - ng_vlan.h, 1679
- NGM_VLAN_COOKIE
 - ng_vlan.h, 1679
- NGM_VLAN_DEL_FILTER
 - ng_vlan.h, 1679
- NGM_VLAN_GET_TABLE
 - ng_vlan.h, 1679
- NGM_XXX_COOKIE
 - ng_sample.h, 1579
- NGM_XXX_GET_STATUS
 - ng_sample.h, 1579
- NGM_XXX_SET_FLAG
 - ng_sample.h, 1579
- NGNAT_ADDR_DEFINED
 - ng_nat.c, 1432
- NGNAT_READY
 - ng_nat.c, 1432
- ngnf_flows, 448
 - entries, 448
 - last, 448
 - nentries, 448
- ngpcb, 449
 - LIST_ENTRY, 450

- ng_socket, 450
- sockdata, 450
- type, 450
- ngpdg_recvspace
 - ng_socket.c, 1591
- ngpdg_sendspace
 - ng_socket.c, 1591
- ngpppoe_init_data, 451
 - data, 451
 - data_len, 451
 - hook, 451
- ngpppoe_init_data_state_type
 - ng_pppoe.c, 1528
- ngpppoe_init_data_type_fields
 - ng_pppoe.c, 1529
- ngpppoe_sts, 452
 - hook, 452
- ngpppoeostat, 453
 - packets_in, 453
 - packets_out, 453
- NGQ_RMASK
 - ng_base.c, 1180
- NGQ_WMASK
 - ng_base.c, 1181
- NGQF_DATA
 - netgraph.h, 1148
- NGQF_FN
 - netgraph.h, 1148
- NGQF_MESG
 - netgraph.h, 1148
- NGQF_QMODE
 - netgraph.h, 1148
- NGQF_QREADER
 - netgraph.h, 1149
- NGQF_QWRITER
 - netgraph.h, 1149
- NGQF_READER
 - netgraph.h, 1149
- NGQF_RW
 - netgraph.h, 1149
- NGQF_TYPE
 - netgraph.h, 1149
- NGQF_UNDEF
 - netgraph.h, 1149
- NGQF_WRITER
 - netgraph.h, 1149
- NGQRW_R
 - ng_base.c, 1181
- NGQRW_W
 - ng_base.c, 1181
- NGRESP_SIZE
 - ng_netflow.h, 1112
- ngs_connect
 - ng_socket.c, 1588, 1591
- ngs_constructor
 - ng_socket.c, 1588, 1591
- ngs_disconnect
 - ng_socket.c, 1588, 1591
- NGS_FLAG_NOLINGER
 - ng_socketvar.h, 1595
- ngs_mod_event
 - ng_socket.c, 1588
- ngs_newhook
 - ng_socket.c, 1588, 1591
- ngs_rcvdata
 - ng_socket.c, 1589, 1591
- ngs_rcvmsg
 - ng_socket.c, 1589, 1591
- ngs_shutdown
 - ng_socket.c, 1589, 1592
- ngsock, 454
 - ctlsock, 454
 - datasock, 454
 - error, 455
 - flags, 455
 - mtx, 455
 - node, 455
 - refs, 455
- ngsw
 - ng_socket.c, 1592
- ngt_close
 - ng_tee.c, 1639, 1642
 - ng_tty.c, 1648
- ngt_connect
 - ng_tty.c, 1649, 1651
- ngt_constructor
 - ng_tee.c, 1639, 1642
 - ng_tty.c, 1649, 1651
- ngt_disc
 - ng_tty.c, 1651
- ngt_disconnect
 - ng_tee.c, 1639, 1642
 - ng_tty.c, 1649, 1652
- NGT_HIWATER
 - ng_tty.c, 1648
- ngt_input
 - ng_tty.c, 1649
- ngt_ldisc
 - ng_tty.c, 1652
- ngt_mod_event
 - ng_tty.c, 1649
- ngt_newhook
 - ng_tee.c, 1640, 1642
 - ng_tty.c, 1649, 1652
- ngt_open
 - ng_tty.c, 1649
- ngt_rcvdata
 - ng_tee.c, 1640, 1642

- ng_tty.c, 1650, 1652
- ngt_rcvmsg
 - ng_tty.c, 1650, 1652
- ngt_read
 - ng_tty.c, 1650
- ngt_sc, 456
 - chand, 456
 - flags, 456
 - hook, 457
 - hotchar, 457
 - m, 457
 - node, 457
 - outq, 457
 - tp, 457
- ngt_shutdown
 - ng_tty.c, 1650, 1652
- ngt_start
 - ng_tty.c, 1650
- ngt_timeout
 - ng_tty.c, 1651
- ngt_tioctl
 - ng_tty.c, 1651
- ngt_unit
 - ng_tty.c, 1652
- ngt_write
 - ng_tty.c, 1651
- NGTLOCK
 - ng_tty.c, 1648
- NGTUNLOCK
 - ng_tty.c, 1648
- nguni_freemem
 - ng_uni.c, 649
- nguni_unilist_mtx
 - ng_uni.c, 649
- nguni_usedmem
 - ng_uni.c, 650
- ngvcc, 458
 - flags, 458
 - hook, 458
 - vci, 458
 - vpi, 459
- ngxxxstat, 460
 - packets_in, 460
 - packets_out, 460
- NI
 - ng_h4.c, 680
- NLPID_CLNP
 - ng_rfc1490.c, 1566
- NLPID_ESIS
 - ng_rfc1490.c, 1566
- NLPID_IP
 - ng_rfc1490.c, 1566
- NLPID_ISIS
 - ng_rfc1490.c, 1566
- NLPID_PPP
 - ng_rfc1490.c, 1566
- NLPID_Q933
 - ng_rfc1490.c, 1566
- NLPID_SNAP
 - ng_rfc1490.c, 1567
- nmembers
 - ng_l2cap_l2ca_grp_get_members_op, 298
- node
 - bt3c_softc, 56
 - cchook, 60
 - ccnode, 62
 - cisco_priv, 67
 - ETF, 73
 - frmrel_softc, 84
 - netflow, 105
 - ng_async_private, 118
 - ng_bpf_hookinfo, 125
 - ng_bridge_private, 140
 - ng_eiface_private, 197
 - ng_fec_private, 205
 - ng_gif_demux_private, 206
 - ng_h4_info, 209
 - ng_hci_unit, 237
 - ng_iface_private, 249
 - ng_ksocket_private, 258
 - ng_l2cap, 263
 - ng_l2tp_private, 312
 - ng_nat_priv, 329
 - ng_one2many_private, 345
 - ng_split_private, 386
 - ng_sppp_private, 388
 - ngd_private, 418
 - nglmi_softc, 421
 - ngm_atm_acr_change, 424
 - ngm_atm_if_change, 430
 - ngm_atm_vcc_change, 432
 - ngsock, 455
 - ngt_sc, 457
 - PPPoE, 464
 - privdata, 478
 - vatmpif_unit, 524
 - XXX, 528
- node_p
 - netgraph.h, 1151
- nodeid
 - ng_ksocket_accept, 256
- nodeinfo, 461
 - hooklist, 88
 - hooks, 461
 - id, 461
 - linkinfo, 102

- name, 461
- namelist, 103
- type, 461
- nomatch_hook
 - ETF, 73
 - priv_p, 473
- nominal_in
 - ngm_bandwidth, 433
- nominal_out
 - ngm_bandwidth, 433
- NOPROTO
 - ng_lmi.c, 1396
- notreviewed.dox, 531
- nports
 - ngm_ccatm_portlist, 437
- nr
 - l2tp_seq, 99
 - ng_l2tp_hook_private, 310
 - ng_l2tp_seq_config, 313
- NREC_AT_ONCE
 - ng_netflow.h, 1112
- nrm
 - ngm_atm_cpcs_init, 427
- ns
 - l2tp_seq, 100
 - ng_l2tp_hook_private, 310
 - ng_l2tp_seq_config, 313
- num_acl_pkt
 - __attribute__, 48
- num_channels
 - ng_btsocket_l2cap_raw_chan_list, 169
 - ng_l2cap_node_chan_list_ep, 305
- num_cmd_pkts
 - __attribute__, 49
- num_compl_pkts
 - ng_hci_evt.c, 735
- num_con_handles
 - __attribute__, 49
- num_connections
 - ng_btsocket_hci_raw_con_list, 147
 - ng_btsocket_l2cap_raw_con_list, 170
 - ng_hci_node_con_list_ep, 228
 - ng_l2cap_node_con_list_ep, 307
- num_entries
 - ng_btsocket_hci_raw_node_neighbor_cache, 155
 - ng_hci_node_get_neighbor_cache_ep, 229
- NUM_FAMILIES
 - ng_gif_demux.c, 1319
 - ng_iface.c, 1338
- num_iac
 - __attribute__, 49
- num_keys
 - __attribute__, 49
- num_keys_deleted
 - __attribute__, 49
- num_keys_read
 - __attribute__, 49
- num_keys_write
 - __attribute__, 49
- num_keys_written
 - __attribute__, 49
- num_names
 - ng_btsocket_hci_raw_node_list_names, 154
- num_pkts
 - ng_hci_node_up_ep, 233
 - ng_l2cap, 263
- num_responses
 - __attribute__, 49
- num_sco_pkt
 - __attribute__, 49
- numActiveLinks
 - ng_ppp_private, 366
- numActiveMany
 - ng_one2many_private, 345
- numBuckets
 - ng_bridge_private, 140
- numHosts
 - ng_bridge_host_ary, 133
 - ng_bridge_private, 140
- numLinks
 - ng_bridge_private, 140
- numnames
 - namelist, 103
- numnodes
 - typeinfo, 507
- NUMTAGS
 - ng_pppoe.c, 1520
- numtags
 - sess_neg, 495
- numtypes
 - typelist, 508
- o_ifx
 - netflow_v5_record, 114
- Octets
 - ng_tcpmss_hookstat, 400
- octets
 - netflow_v1_record, 108
 - netflow_v5_record, 114
 - ng_hole_hookstat, 245
- oerrors
 - ng_bt3c_node_stat_ep, 143
 - ng_h4_node_stat_ep, 211
 - ng_ubt_node_stat_ep, 407
- OFFSETOF
 - ng_bpf.c, 1206
 - ng_ksocket.c, 1362

- ng_pppoe.c, 1520
- offlow
 - ng_btsocket_l2cap_pcb, 165
 - ng_l2cap_chan, 267
 - ng_l2cap_l2ca_cfg_ip, 278
 - ng_l2cap_l2ca_cfg_op, 279
- omtu
 - ng_btsocket_l2cap_pcb, 165
 - ng_l2cap_chan, 267
 - ng_l2cap_l2ca_cfg_ind_ip, 277
 - ng_l2cap_l2ca_cfg_rsp_ip, 280
 - ng_l2cap_node_chan_ep, 303
- one
 - ng_one2many_private, 345
- op
 - ccatm_op, 58
- OP_PENDING
 - ng_base.c, 1181
- opcode
 - __attribute__, 49
- option_mask
 - ngm_uni_config_mask, 444
- orphan
 - private, 476
- orphans
 - priv, 471
- OUICMP
 - ng_rfc1490.c, 1567
- ourhook
 - linkinfo, 102
 - ngm_connect, 438
 - ngm_mkpeer, 439
 - ngm_rmhook, 443
- out
 - ng_nat_priv, 329
 - ng_netflow_iface, 330
 - ng_split_private, 387
 - ng_tag_hookinfo, 396
- out_dropped
 - stats, 506
- out_errors
 - ngm_atm_stats, 431
 - priv, 471
- out_ifx
 - netflow_v1_record, 108
- out_packets
 - ngm_atm_stats, 431
 - priv, 471
 - stats, 506
- out_tag_cookie
 - ng_tag_hookinfo, 396
- out_tag_data
 - ng_tag_hookinfo, 396
- out_tag_id
 - ng_tag_hookinfo, 396
- out_tag_len
 - ng_tag_hookinfo, 396
- outbuf
 - ng_deflate_private, 194
 - ng_pred1_private, 377
- outFrames
 - ng_source_stats, 384
 - ng_tee_hookstat, 401
- outHook
 - hpriv_p, 89
 - ng_tcpmss_config, 399
- OutOctets
 - ng_deflate_stats, 196
 - ng_pred1_stats, 378
- outOctets
 - ng_source_stats, 384
 - ng_tee_hookstat, 401
- output
 - priv, 471
 - privdata, 478
- output_ifp
 - privdata, 478
- outq
 - bt3c_softc, 56
 - ng_btsocket_rfcomm_session, 187
 - ng_h4_info, 209
 - ngt_sc, 457
- OutSeq
 - ng_atmpif_link_status, 124
 - ng_vatmpif_hook, 411
- pack_buf
 - ng_ccatm.c, 559
- packed
 - ng_btsocket_rfcomm.h, 842
- packet, 462
 - bytes, 462
 - pkt_header, 462
- packet_mask
 - ng_btsocket_hci_raw_node_packet_mask, 156
 - ng_hci_unit, 237
- Packets
 - ng_tcpmss_hookstat, 400
- packets
 - flow_entry_data, 78
 - netflow_v1_record, 108
 - netflow_v5_record, 114
 - privdata, 478
- packets_in
 - ETF, 73
 - ng_etfstat, 200
 - ngpppoestat, 453
 - ngxxxstat, 460

- PPPoE, 464
- XXX, 528
- packets_out
 - ETF, 73
 - ng_etfstat, 200
 - ngpppoestat, 453
 - ngxxxstat, 460
 - PPPoE, 464
 - XXX, 528
- pad
 - netflow_v5_header, 111
- pad1
 - netflow_v1_record, 108
 - netflow_v5_record, 114
- pad2
 - netflow_v1_record, 108
 - netflow_v5_record, 115
- pad3
 - netflow_v1_record, 108
- PADI_CODE
 - ng_pppoe.h, 1534
- PADO_CODE
 - ng_pppoe.h, 1534
- PADR_CODE
 - ng_pppoe.h, 1534
- PADS_CODE
 - ng_pppoe.h, 1534
- PADT_CODE
 - ng_pppoe.h, 1534
- page_scan_interval
 - __attribute__, 50
- page_scan_mode
 - __attribute__, 50
 - ng_hci_neighbor, 222
 - ng_hci_node_neighbor_cache_entry_ep, 230
- page_scan_mode_change
 - ng_hci_evt.c, 735
- page_scan_period_mode
 - __attribute__, 50
- page_scan_rep_mode
 - __attribute__, 50
 - ng_hci_neighbor, 223
 - ng_hci_node_neighbor_cache_entry_ep, 230
- page_scan_rep_mode_change
 - ng_hci_evt.c, 735
- page_scan_window
 - __attribute__, 50
- par1
 - cisco_packet, 64
- par2
 - cisco_packet, 64
- param
 - ng_sscfu_getdefparam, 390
 - ng_sscop_setparam, 391
- param_mask
 - rfcomm_mcc_rpn, 488
- parse
 - ng_parse_type, 350
- PARTY_ALLOC
 - ng_uni_cust.h, 653
- PARTY_FREE
 - ng_uni_cust.h, 654
- path
 - ngm_connect, 438
- pcb_mtx
 - ng_btsocket_hci_raw_pcb, 161
 - ng_btsocket_l2cap_pcb, 165
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_btsocket_rfcomm_pcb, 184
- pckts_recv
 - ng_bt3c_node_stat_ep, 143
 - ng_h4_node_stat_ep, 211
 - ng_ubt_node_stat_ep, 407
- pckts_sent
 - ng_bt3c_node_stat_ep, 143
 - ng_h4_node_stat_ep, 211
 - ng_ubt_node_stat_ep, 407
- pcr
 - ng_vatmpif_config, 409
 - ngm_atm_config, 425
 - ngm_atm_cpcs_init, 427
- peak_bandwidth
 - __attribute__, 50
 - ng_hci_lp_qos_req_ep, 220
- peer_id
 - ng_l2tp_config, 308
 - ng_l2tp_sess_config, 314
- peer_win
 - ng_l2tp_config, 308
- peerCid
 - ng_pptpgre_conf, 369
- peerhook
 - linkinfo, 102
 - ngm_connect, 438
 - ngm_mkpeer, 439
- peerPpd
 - ng_pptpgre_conf, 369
- pending
 - ng_hci_node_con_ep, 226
 - ng_hci_unit_con, 243
 - ng_l2cap_con, 274
 - ng_l2cap_node_con_ep, 306
- ph
 - pppoe_full_hdr, 465
- pin
 - __attribute__, 50
- pin_size
 - __attribute__, 50

- pin_type
 - __attribute__, 50
- pkt
 - sess_neg, 495
- pkt_hdr
 - sess_con, 493
- pkt_header
 - packet, 462
- pkt_size
 - ng_hci_node_up_ep, 233
 - ng_l2cap, 263
- pkt_type
 - __attribute__, 50
- pointer
 - uniq, 521
- policy_mask
 - ng_btsocket_hci_raw_node_link_policy_mask, 153
- poll_count
 - nglmi_softc, 421
- poll_state
 - nglmi_softc, 421
- popt_mask
 - ngm_uni_config_mask, 444
- port
 - ngm_ccatm_addr_req, 434
 - ngm_ccatm_port, 436
- ports
 - flow_rec, 82
 - ngm_ccatm_portlist, 437
- PPP
 - ng_rfc1490_private, 383
- PPP_GOODFCS
 - ng_pred1.c, 1556
- PPP_INITFCS
 - ng_pred1.c, 1556
- PPPoE, 463
- PPPoE
 - debug_hook, 463
 - eh, 463
 - ethernet_hook, 463
 - flags, 464
 - node, 464
 - packets_in, 464
 - packets_out, 464
- pppoe_broadcast_padi
 - ng_pppoe.c, 1525
- PPPOE_CONNECTED
 - ng_pppoe.c, 1520
- PPPOE_DEAD
 - ng_pppoe.c, 1521
- pppoe_find_svc
 - ng_pppoe.c, 1525
- pppoe_findsession
 - ng_pppoe.c, 1525
- pppoe_finduniq
 - ng_pppoe.c, 1525
- pppoe_full_hdr, 465
 - eh, 465
 - ph, 465
- pppoe_hdr, 466
 - code, 466
 - length, 466
 - sid, 466
 - tag, 466
 - type, 466
 - ver, 467
- PPPOE_INITIAL_TIMEOUT
 - ng_pppoe.h, 1534
- PPPOE_LISTENING
 - ng_pppoe.c, 1520
- pppoe_match_svc
 - ng_pppoe.c, 1525
- PPPOE_NEWCONNECTED
 - ng_pppoe.c, 1520
- PPPOE_NUM_SESSIONS
 - ng_pppoe.h, 1534
- PPPOE_OFFER_TIMEOUT
 - ng_pppoe.h, 1534
- PPPOE_PRIMED
 - ng_pppoe.c, 1520
- pppoe_send_event
 - ng_pppoe.c, 1525
- PPPOE_SERVICE_NAME_SIZE
 - ng_pppoe.h, 1534
- PPPOE_SINIT
 - ng_pppoe.c, 1520
- PPPOE_SNONE
 - ng_pppoe.c, 1520
- PPPOE_SOFFER
 - ng_pppoe.c, 1520
- PPPOE_SREQ
 - ng_pppoe.c, 1520
- pppoe_start
 - ng_pppoe.c, 1526
- pppoe_tag, 468
 - tag_data, 468
 - tag_len, 468
 - tag_type, 468
- pppoe_ticker
 - ng_pppoe.c, 1526
- PPPOE_TIMEOUT_LIMIT
 - ng_pppoe.h, 1535
- PPTP_ACK_ALPHA
 - ng_pptpgre.c, 1542
- PPTP_ACK_BETA
 - ng_pptpgre.c, 1542
- PPTP_ACK_CHI

- ng_pptpgre.c, 1542
- PPTP_ACK_DELTA
 - ng_pptpgre.c, 1542
- PPTP_GRE_PROTO
 - ng_pptpgre.c, 1542
- PPTP_INIT_MASK
 - ng_pptpgre.c, 1542
- PPTP_INIT_VALUE
 - ng_pptpgre.c, 1542
- PPTP_MAX_ACK_DELAY
 - ng_pptpgre.c, 1542
- PPTP_MAX_PAYLOAD
 - ng_pptpgre.c, 1542
- PPTP_MAX_TIMEOUT
 - ng_pptpgre.c, 1543
- PPTP_MIN_ACK_DELAY
 - ng_pptpgre.c, 1543
- PPTP_MIN_RTT
 - ng_pptpgre.c, 1543
- PPTP_MIN_TIMEOUT
 - ng_pptpgre.c, 1543
- PPTP_SEQ_DIFF
 - ng_pptpgre.c, 1543
- PPTP_TIME_SCALE
 - ng_pptpgre.c, 1543
- PPTP_XMIT_WIN
 - ng_pptpgre.c, 1543
- pptptime_t
 - ng_pptpgre.c, 1544
- PRED1_BUF_SIZE
 - ng_pred1.c, 1556
- PRED1_HDRLEN
 - ng_pred1.c, 1556
- PRED1_TABLE_SIZE
 - ng_pred1.c, 1556
- Pred1Compress
 - ng_pred1.c, 1559
- Pred1Decompress
 - ng_pred1.c, 1559
- Pred1Init
 - ng_pred1.c, 1559
- Pred1SyncTable
 - ng_pred1.c, 1559
- priority
 - ng_tag_prio, 398
 - rfcomm_mcc_pn, 486
- priv, 469
 - enabled, 470
 - flow, 470
 - ifp, 470
 - in_errors, 470
 - in_packets, 470
 - input, 470
 - lower, 470
 - manage, 470
 - orphans, 471
 - out_errors, 471
 - out_packets, 471
 - output, 471
 - sscf, 471
 - sscop, 471
 - stats, 471
 - uni, 471
 - upper, 472
- priv_p, 473
 - downstream_hook, 473
 - hashtable, 473
 - nent, 473
 - ng_atmpif_var.h, 549
 - ng_bridge.c, 1217
 - ng_deflate.c, 1242
 - ng_device.c, 1252
 - ng_eiface.c, 1263
 - ng_ether.c, 1280
 - ng_fec.c, 1291
 - ng_gif.c, 1310
 - ng_gif_demux.c, 1320
 - ng_iface.c, 1339
 - ng_ksocket.c, 1363
 - ng_l2tp.c, 1379
 - ng_mppc.c, 1423
 - ng_nat.c, 1432
 - ng_netflow.h, 1113
 - ng_one2many.c, 1438
 - ng_ppp.c, 1492
 - ng_pppoe.c, 1520
 - ng_pptpgre.c, 1544
 - ng_pred1.c, 1556
 - ng_rfc1490.c, 1567
 - ng_split.c, 1608
 - ng_sppp.c, 1614
 - ng_UI.c, 1657
 - ng_vjc.c, 1663
 - nomatch_hook, 473
- private, 475
 - autoSrcAddr, 475
 - flags, 475
 - hwassist, 475
 - ifp, 476
 - lower, 476
 - lowerOrphan, 476
 - ng_parse_type, 351
 - orphan, 476
 - promisc, 476
 - upper, 476
- privdata, 477
 - input, 478
 - intr_ch, 478

- left, [478](#)
- left2right, [478](#)
- node, [478](#)
- output, [478](#)
- output_ifp, [478](#)
- packets, [478](#)
- queueOctets, [478](#)
- right, [478](#)
- right2left, [478](#)
- snd_queue, [478](#)
- stats, [479](#)
- process_hc_baseband_params
 - ng_hci_cmds.c, [722](#)
- process_info_params
 - ng_hci_cmds.c, [723](#)
- process_link_control_params
 - ng_hci_cmds.c, [723](#)
- process_link_control_status
 - ng_hci_cmds.c, [724](#)
- process_link_policy_params
 - ng_hci_cmds.c, [724](#)
- process_link_policy_status
 - ng_hci_cmds.c, [724](#)
- process_status_params
 - ng_hci_cmds.c, [724](#)
- process_testing_params
 - ng_hci_cmds.c, [725](#)
- prog
 - ng_bpf_hookinfo, [125](#)
- promisc
 - private, [476](#)
- prot
 - flow_rec, [82](#)
 - netflow_v1_record, [108](#)
 - netflow_v5_record, [115](#)
- PROT_ATALK
 - ng_ppp.c, [1491](#)
- PROT_COMPD
 - ng_deflate.c, [1242](#)
 - ng_ppp.c, [1491](#)
- PROT_COMPRESSABLE
 - ng_ppp.c, [1491](#)
- PROT_CRYPTD
 - ng_ppp.c, [1491](#)
- PROT_IP
 - ng_ppp.c, [1491](#)
- PROT_IPV6
 - ng_ppp.c, [1491](#)
- PROT_IPX
 - ng_ppp.c, [1492](#)
- PROT_LCP
 - ng_ppp.c, [1492](#)
- PROT_MP
 - ng_ppp.c, [1492](#)
- PROT_VALID
 - ng_ppp.c, [1492](#)
- PROT_VJCOMP
 - ng_ppp.c, [1492](#)
- PROT_VJUNCOMP
 - ng_ppp.c, [1492](#)
- proto
 - greheader, [86](#)
 - nglmistat, [423](#)
- protocol
 - cisco_header, [63](#)
- protoent, [480](#)
 - af, [480](#)
 - hook, [480](#)
- protoID
 - nglmi_softc, [421](#)
- protoname
 - nglmi_softc, [421](#)
- psm
 - __attribute__, [50](#)
 - ng_btsocket_l2cap_pcb, [165](#)
 - ng_l2cap_chan, [267](#)
 - ng_l2cap_l2ca_con_ind_ip, [282](#)
 - ng_l2cap_l2ca_con_ip, [283](#)
 - ng_l2cap_l2ca_enable_clt_ip, [289](#)
 - ng_l2cap_l2ca_grp_create_ip, [295](#)
 - ng_l2cap_node_chan_ep, [303](#)
- PTT_AC_COOKIE
 - ng_pppoe.h, [1535](#)
- PTT_AC_NAME
 - ng_pppoe.h, [1535](#)
- PTT_EOL
 - ng_pppoe.h, [1535](#)
- PTT_GEN_ERR
 - ng_pppoe.h, [1535](#)
- PTT_HOST_UNIQ
 - ng_pppoe.h, [1535](#)
- PTT_RELAY_SID
 - ng_pppoe.h, [1535](#)
- PTT_SRV_ERR
 - ng_pppoe.h, [1535](#)
- PTT_SRV_NAME
 - ng_pppoe.h, [1535](#)
- PTT_SYS_ERR
 - ng_pppoe.h, [1536](#)
- PTT_VENDOR
 - ng_pppoe.h, [1536](#)
- q_flags
 - ng_queue, [379](#)
- q_mtx
 - ng_queue, [380](#)
- q_node
 - ng_queue, [380](#)

- qlen
 - ng_bt3c_node_qlen_ep, 142
 - ng_ubt_node_qlen_ep, 406
- qos_setup_compl
 - ng_hci_evt.c, 736
- qos_violation
 - ng_hci_evt.c, 736
- quality
 - __attribute__, 51
- queue
 - ng_bt3c_node_qlen_ep, 142
 - ng_queue, 380
 - ng_ubt_node_qlen_ep, 406
- QUEUE_ACTIVE
 - ng_base.c, 1181
- queue_len
 - ng_hci_node_con_ep, 226
- queue_priority
 - ngm_queue_state, 441
- QUEUED_READER_CAN_PROCEED
 - ng_base.c, 1181
- QUEUED_WRITER_CAN_PROCEED
 - ng_base.c, 1181
- queueFrames
 - ng_source_stats, 384
- queueOctets
 - ng_source_stats, 385
 - privdata, 478
- r
 - flow_entry_data, 78
 - netflow_v5_export_dgram, 110
- r_dport
 - ng_netflow.h, 1112
- r_dst
 - flow_rec, 82
- r_i_ifx
 - ng_netflow.h, 1112
- r_ip_p
 - ng_netflow.h, 1112
- r_misc
 - ng_netflow.h, 1112
- r_ports
 - ng_netflow.h, 1112
- r_sport
 - ng_netflow.h, 1113
- r_src
 - flow_rec, 82
- r_tos
 - ng_netflow.h, 1113
- rack
 - l2tp_seq, 100
 - ng_l2tp_seq_config, 313
- rack_timer
 - l2tp_seq, 100
- rackTimer
 - ng_pptpgre_ackp, 367
- raw
 - ng_hci_unit, 237
- rcs_id
 - netflow.c, 1097
 - ng_netflow.c, 1107
- rcvdata
 - ng_type, 405
- rcvmsg
 - ng_type, 405
- rdf
 - ngm_atm_cpcs_init, 427
- read_all
 - __attribute__, 51
- read_clock_offset_compl
 - ng_hci_evt.c, 736
- read_remote_features_compl
 - ng_hci_evt.c, 737
- READER_INCREMENT
 - ng_base.c, 1181
- READER_MASK
 - ng_base.c, 1182
- readq
 - ngd_private, 418
- reason
 - __attribute__, 51
 - ng_hci_lp_discon_ind_ep, 216
 - ng_hci_lp_discon_req_ep, 217
- recursion
 - greheader, 86
- recv
 - ng_mppc_private, 327
- recvAck
 - ng_pptpgre_private, 372
- recvAckTimeouts
 - ng_pptpgre_stats, 373
- recvBadAcks
 - ng_l2tp_stats, 316
 - ng_pptpgre_stats, 373
- recvBadCID
 - ng_pptpgre_stats, 373
- recvBadGRE
 - ng_pptpgre_stats, 373
- recvBroadcasts
 - ng_bridge_link_stats, 136
- recvDataDrops
 - ng_l2tp_stats, 316
- recvDuplicates
 - ng_l2tp_stats, 316
 - ng_pptpgre_stats, 373
- recvFrames
 - ng_bpf_hookstat, 128

- ng_ppp_link_stat, 361
- recvInvalid
 - ng_bridge_link_stats, 136
 - ng_l2tp_stats, 316
- recvLoneAcks
 - ng_pptpgre_stats, 374
- recvMatchFrames
 - ng_bpf_hookstat, 128
- recvMatchOctets
 - ng_bpf_hookstat, 128
- recvMulticasts
 - ng_bridge_link_stats, 137
- recvOctets
 - ng_bpf_hookstat, 128
 - ng_bridge_link_stats, 137
 - ng_l2tp_session_stats, 315
 - ng_l2tp_stats, 317
 - ng_one2many_link_stats, 343
 - ng_ppp_link_stat, 361
 - ng_pptpgre_stats, 374
- recvOutOfOrder
 - ng_l2tp_stats, 317
 - ng_pptpgre_stats, 374
- recvPackets
 - ng_bridge_link_stats, 137
 - ng_l2tp_session_stats, 315
 - ng_l2tp_stats, 317
 - ng_one2many_link_stats, 343
 - ng_pptpgre_stats, 374
- recvRunts
 - ng_bridge_link_stats, 137
 - ng_l2tp_stats, 317
 - ng_pptpgre_stats, 374
- recvSeq
 - ng_pptpgre_private, 372
- recvShortSeq
 - ng_ppp_bund_conf, 353
- recvUnknown
 - ng_bridge_link_stats, 137
- recvUnknownSID
 - ng_l2tp_stats, 317
- recvWin
 - ng_pptpgre_conf, 370
- recvWrongTunnel
 - ng_l2tp_stats, 317
- recvZLBs
 - ng_l2tp_stats, 317
- refcnt
 - ng_l2cap_con, 274
- refs
 - ng_type, 405
 - ngsock, 455
- rel
 - cisco_packet, 64
- remote
 - ng_l2cap_con, 274
 - ng_l2cap_node_chan_ep, 303
 - ng_l2cap_node_con_ep, 306
- remote_seq
 - cisco_priv, 67
 - frmrel_softc, 84
 - nglmi_softc, 421
- reserved
 - netflow_v1_record, 108
 - ng_btsocket_rfcomm_fc_info, 180
 - ng_hci_node_con_ep, 227
 - ng_hci_node_up_ep, 233
- respType
 - ng_cmdlist, 191
- result
 - __attribute__, 51
 - ng_btsocket_l2cap_raw_get_info, 171
 - ng_btsocket_l2cap_raw_ping, 177
 - ng_l2cap_l2ca_cfg_op, 279
 - ng_l2cap_l2ca_cfg_rsp_op, 281
 - ng_l2cap_l2ca_con_op, 284
 - ng_l2cap_l2ca_con_rsp_ip, 285
 - ng_l2cap_l2ca_con_rsp_op, 286
 - ng_l2cap_l2ca_discon_op, 288
 - ng_l2cap_l2ca_get_info_op, 291
 - ng_l2cap_l2ca_grp_add_member_op, 293
 - ng_l2cap_l2ca_grp_get_members_op, 298
 - ng_l2cap_l2ca_ping_op, 300
 - ng_l2cap_l2ca_write_op, 302
- return_export_dgram
 - netflow.c, 1096
- rexit_max
 - ng_l2tp_config, 308
- rexit_max_to
 - ng_l2tp_config, 308
- rexmits
 - l2tp_seq, 100
- rfcomm_bdaddr
 - sockaddr_rfcomm, 500
- RFCOMM_CHANNEL
 - ng_btsocket_rfcomm.h, 829
- rfcomm_channel
 - sockaddr_rfcomm, 500
- rfcomm_cmd_hdr, 481
 - address, 481
 - control, 481
 - fcs, 481
 - length, 481
- RFCOMM_CR
 - ng_btsocket_rfcomm.h, 829
- RFCOMM_DEFAULT_CREDITS
 - ng_btsocket_rfcomm.h, 829
- RFCOMM_DEFAULT_MTU

- ng_btsocket_rfcomm.h, 829
- RFCOMM_DIRECTION
 - ng_btsocket_rfcomm.h, 830
- RFCOMM_DLCI
 - ng_btsocket_rfcomm.h, 830
- RFCOMM_EA
 - ng_btsocket_rfcomm.h, 830
- rfcomm_family
 - sockaddr_rfcomm, 500
- RFCOMM_FRAME_DISC
 - ng_btsocket_rfcomm.h, 830
- RFCOMM_FRAME_DM
 - ng_btsocket_rfcomm.h, 830
- rfcomm_frame_hdr, 482
 - address, 482
 - control, 482
 - length, 482
- RFCOMM_FRAME_SABM
 - ng_btsocket_rfcomm.h, 830
- RFCOMM_FRAME_UA
 - ng_btsocket_rfcomm.h, 830
- RFCOMM_FRAME_UIH
 - ng_btsocket_rfcomm.h, 830
- rfcomm_len
 - sockaddr_rfcomm, 500
- RFCOMM_MAX_CREDITS
 - ng_btsocket_rfcomm.h, 831
- RFCOMM_MAX_MTU
 - ng_btsocket_rfcomm.h, 831
- RFCOMM_MCC_FCOFF
 - ng_btsocket_rfcomm.h, 831
- RFCOMM_MCC_FCON
 - ng_btsocket_rfcomm.h, 831
- rfcomm_mcc_hdr, 483
 - length, 483
 - type, 483
- RFCOMM_MCC_LENGTH
 - ng_btsocket_rfcomm.h, 831
- RFCOMM_MCC_MSC
 - ng_btsocket_rfcomm.h, 831
- rfcomm_mcc_msc, 484
 - address, 484
 - modem, 484
- RFCOMM_MCC_NSC
 - ng_btsocket_rfcomm.h, 831
- RFCOMM_MCC_PN
 - ng_btsocket_rfcomm.h, 831
- rfcomm_mcc_pn, 485
 - ack_timer, 485
 - credits, 485
 - dhci, 485
 - flow_control, 485
 - max_retrans, 485
 - mtu, 485
 - priority, 486
- RFCOMM_MCC_RLS
 - ng_btsocket_rfcomm.h, 832
- rfcomm_mcc_rls, 487
 - address, 487
 - status, 487
- RFCOMM_MCC_RPN
 - ng_btsocket_rfcomm.h, 832
- rfcomm_mcc_rpn, 488
 - bit_rate, 488
 - dhci, 488
 - flow_control, 488
 - line_settings, 488
 - param_mask, 488
 - xoff_char, 488
 - xon_char, 489
- RFCOMM_MCC_TEST
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MCC_TYPE
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MKADDRESS
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MKCONTROL
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MKDLCI
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MKLEN16
 - ng_btsocket_rfcomm.h, 832
- RFCOMM_MKLEN8
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MKMCC_TYPE
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MKRPN_LINE_SETTINGS
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MODEM_DV
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MODEM_FC
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MODEM_IC
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MODEM_RTC
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_MODEM_RTR
 - ng_btsocket_rfcomm.h, 833
- RFCOMM_PF
 - ng_btsocket_rfcomm.h, 834
- RFCOMM_RPN_BR_115200
 - ng_btsocket_rfcomm.h, 834
- RFCOMM_RPN_BR_19200
 - ng_btsocket_rfcomm.h, 834
- RFCOMM_RPN_BR_230400
 - ng_btsocket_rfcomm.h, 834
- RFCOMM_RPN_BR_2400
 - ng_btsocket_rfcomm.h, 834

- RFCOMM_RPN_BR_38400
 - ng_btsocket_rfcomm.h, [834](#)
- RFCOMM_RPN_BR_4800
 - ng_btsocket_rfcomm.h, [834](#)
- RFCOMM_RPN_BR_57600
 - ng_btsocket_rfcomm.h, [834](#)
- RFCOMM_RPN_BR_7200
 - ng_btsocket_rfcomm.h, [834](#)
- RFCOMM_RPN_BR_9600
 - ng_btsocket_rfcomm.h, [834](#)
- RFCOMM_RPN_DATA_5
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_DATA_6
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_DATA_7
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_DATA_8
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_DATA_BITS
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_FLOW_NONE
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_PARITY
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_PARITY_EVEN
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_PARITY_MARK
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_PARITY_NONE
 - ng_btsocket_rfcomm.h, [835](#)
- RFCOMM_RPN_PARITY_ODD
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PARITY_SPACE
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_ALL
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_BITRATE
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_DATA
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_FLOW
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_PARITY
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_PARITY_TYPE
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_STOP
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_XOFF
 - ng_btsocket_rfcomm.h, [836](#)
- RFCOMM_RPN_PM_XON
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_RPN_STOP_1
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_RPN_STOP_15
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_RPN_STOP_BITS
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_RPN_XOFF_CHAR
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_RPN_XON_CHAR
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_SRVCHANNEL
 - ng_btsocket_rfcomm.h, [837](#)
- RFCOMM_TYPE
 - ng_btsocket_rfcomm.h, [837](#)
- rif
 - ngm_atm_cpcs_init, [427](#)
- right
 - ng_tee_stats, [402](#)
 - privdata, [478](#)
- right2left
 - ng_tee_stats, [402](#)
 - privdata, [478](#)
- rmodem
 - ng_btsocket_rfcomm_fc_info, [180](#)
 - ng_btsocket_rfcomm_pcb, [184](#)
- rmtu
 - ngm_atm_cpcs_init, [427](#)
- role
 - __attribute__, [51](#)
 - ng_hci_node_con_ep, [227](#)
 - ng_hci_unit_con, [243](#)
- role_change
 - ng_hci_evnt.c, [737](#)
- role_switch
 - ng_btsocket_hci_raw_node_role_switch, [157](#)
 - ng_hci_unit, [237](#)
- rseq
 - ng_ppp_mp_state, [363](#)
- rsi
 - __attribute__, [51](#)
- rt
 - ng_btsocket_l2cap_pcb, [166](#)
 - ng_btsocket_l2cap_raw_pcb, [175](#)
- rtt
 - ng_pptpgre_ackp, [367](#)
- rule
 - ng_ipfw_tag, [252](#)
- rulenum
 - ng_ipfw_hook_priv, [251](#)
- running
 - ngm_atm_if_change, [430](#)
- runts
 - ng_ppp_link_stat, [361](#)
- rx_cred
 - ng_btsocket_rfcomm_fc_info, [180](#)
 - ng_btsocket_rfcomm_pcb, [184](#)

- rx_pkt
 - ng_l2cap_con, 274
- rx_pkt_len
 - ng_l2cap_con, 274
- s_port
 - flow_rec, 82
 - netflow_v1_record, 108
 - netflow_v5_record, 115
- sa
 - sa_tag, 490
- SA_SIZE
 - ng_iface.c, 1263
- sa_tag, 490
 - id, 490
 - sa, 490
 - tag, 490
- sackTimer
 - ng_pptpgre_ackp, 367
- SADATA_OFFSET
 - ng_ksocket.c, 1362
- SAFETY_BARRIER
 - ng_base.c, 1182
- SAVE_LINE
 - netgraph.h, 1149
- sbuf
 - ng_async_private, 118
- sc_aclq
 - ubt_softc, 510
- sc_bulk_in_buffer
 - ubt_softc, 510
- sc_bulk_in_ep
 - ubt_softc, 510
- sc_bulk_in_pipe
 - ubt_softc, 510
- sc_bulk_in_xfer
 - ubt_softc, 510
- sc_bulk_out_buffer
 - ubt_softc, 511
- sc_bulk_out_dev
 - ubtbcmfw_softc, 516
- sc_bulk_out_ep
 - ubt_softc, 511
- sc_bulk_out_pipe
 - ubt_softc, 511
 - ubtbcmfw_softc, 516
- sc_bulk_out_xfer
 - ubt_softc, 511
- sc_cmdq
 - ubt_softc, 511
- sc_ctrl_buffer
 - ubt_softc, 511
- sc_ctrl_dev
 - ubtbcmfw_softc, 516
- sc_ctrl_xfer
 - ubt_softc, 511
- sc_debug
 - ubt_softc, 511
- sc_dev
 - ubt_softc, 511
 - ubtbcmfw_softc, 516
- sc_dying
 - ubtbcmfw_softc, 516
- sc_flags
 - ubt_softc, 512
 - ubtbcmfw_softc, 516
- sc_hook
 - ubt_softc, 512
- sc_iface0
 - ubt_softc, 512
- sc_iface1
 - ubt_softc, 512
- sc_intr_buffer
 - ubt_softc, 512
- sc_intr_ep
 - ubt_softc, 512
- sc_intr_in_dev
 - ubtbcmfw_softc, 517
- sc_intr_in_pipe
 - ubtbcmfw_softc, 517
- sc_intr_pipe
 - ubt_softc, 512
- sc_intr_xfer
 - ubt_softc, 512
- sc_isoc_in_buffer
 - ubt_softc, 513
- sc_isoc_in_ep
 - ubt_softc, 513
- sc_isoc_in_frlen
 - ubt_softc, 513
- sc_isoc_in_pipe
 - ubt_softc, 513
- sc_isoc_in_xfer
 - ubt_softc, 513
- sc_isoc_nframes
 - ubt_softc, 513
- sc_isoc_out_buffer
 - ubt_softc, 513
- sc_isoc_out_ep
 - ubt_softc, 513
- sc_isoc_out_frlen
 - ubt_softc, 513
- sc_isoc_out_pipe
 - ubt_softc, 514
- sc_isoc_out_xfer
 - ubt_softc, 514
- sc_isoc_size
 - ubt_softc, 514

- sc_node
 - ubt_softc, 514
- sc_p
 - ng_async.c, 1160
 - ng_cisco.c, 1231
 - ng_frame_relay.c, 1303
 - ng_lmi.c, 1398
 - ng_source.c, 1599
 - ng_tee.c, 1639
 - ng_tty.c, 1648
- sc_refcnt
 - ubtbcmfw_softc, 517
- sc_scoq
 - ubt_softc, 514
- sc_stat
 - ubt_softc, 514
- sc_udev
 - ubt_softc, 514
 - ubtbcmfw_softc, 517
- scan_enable
 - __attribute__, 51
- scan_tags
 - ng_pppoe.c, 1527
- SCF_ANNEX_A
 - ng_lmi.c, 1396
- SCF_ANNEX_D
 - ng_lmi.c, 1396
- SCF_AUTO
 - ng_lmi.c, 1396
- SCF_CONNECTED
 - ng_lmi.c, 1397
- SCF_FIXED
 - ng_lmi.c, 1397
- SCF_GROUP4
 - ng_lmi.c, 1397
- SCF_LMITYPE
 - ng_lmi.c, 1397
- SCF_NOLMI
 - ng_lmi.c, 1397
- scid
 - __attribute__, 51
 - ng_l2cap_chan, 267
 - ng_l2cap_cmd_rej_data_t, 271
 - ng_l2cap_node_chan_ep, 303
- SCM_HCI_RAW_DIRECTION
 - ng_btsocket.h, 798
- sco
 - ng_hci_unit, 238
- sco_free
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- sco_pkts
 - ng_hci_node_buffer_ep, 224
 - ng_hci_unit_buff, 239
- sco_recv
 - ng_hci_node_stat_ep, 231
- sco_sent
 - ng_hci_node_stat_ep, 232
- sco_size
 - ng_hci_node_buffer_ep, 225
 - ng_hci_unit_buff, 239
- scr
 - ngm_atm_cpcs_init, 428
- seen
 - nglmistat, 423
- seen_in
 - ngm_bandwidth, 433
- seen_out
 - ngm_bandwidth, 433
- segment, 491
 - mask, 491
 - shift, 491
 - width, 491
- send_acname
 - ng_pppoe.c, 1527
- send_data_packets
 - ng_hci_evt.c, 737
- send_dump
 - ng_ccatm.c, 559
- send_l2cap_cfg_rsp
 - ng_l2cap_evt.c, 942
- send_l2cap_con_rej
 - ng_l2cap_evt.c, 942
- send_l2cap_reject
 - ng_l2cap_evt.c, 943
- send_sessionid
 - ng_pppoe.c, 1527
- seq
 - ng_l2tp_private, 312
 - ng_ppp_frag, 355
 - ng_ppp_link, 358
 - vatmpif_header, 522
- seq_retries
 - nglmi_softc, 422
- seqnum
 - ng_deflate_private, 195
 - ng_predl_private, 377
- seqRetries
 - cisco_priv, 67
 - ng_cisco_stats, 190
- service
 - sess_neg, 495
- service_len
 - sess_neg, 495
- service_type
 - __attribute__, 51
 - ng_hci_lp_qos_req_ep, 220
- sess_con, 492

- creator, 492
- hook, 492
- neg, 493
- pkt_hdr, 493
- Session_ID, 493
- state, 493
- sess_neg, 494
 - ac_name, 494
 - ac_name_len, 494
 - handle, 494
 - m, 495
 - numtags, 495
 - pkt, 495
 - service, 495
 - service_len, 495
 - tags, 495
 - timeout, 495
- session
 - ng_btsocket_rfcomm_pcb, 184
- Session_ID
 - sess_con, 493
- session_id
 - ng_l2tp_sess_config, 314
- session_mtx
 - ng_btsocket_rfcomm_session, 187
- sessp
 - ng_pppoe.c, 1520
- SET_RETADDR
 - ng_base.c, 1182
- SETLMITYPE
 - ng_lmi.c, 1397
- settings
 - __attribute__, 51
- sg_data
 - sockaddr_ng, 499
- sg_family
 - sockaddr_ng, 499
- sg_len
 - sockaddr_ng, 499
- shift
 - segment, 491
- SHIFTIN
 - ng_frame_relay.c, 1302
- SHIFTOUT
 - ng_frame_relay.c, 1303
- shutdown
 - ng_type, 405
- sid
 - pppoe_hdr, 466
- sig
 - sscfu_arg, 501
 - sscop_arg, 502
 - sscop_marg, 503
 - sscop_merr, 504
 - uni_arg, 518
- SIG_ALLOC
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 635
 - ng_uni_cust.h, 654
- SIG_FREE
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 635
 - ng_uni_cust.h, 654
- SIGNOFF
 - ng_pppoe.c, 1520
- SIGQ_APPEND
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 636
- SIGQ_CLEAR
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 636
- SIGQ_EMPTY
 - ng_sscop_cust.h, 636
- SIGQ_GET
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 636
- SIGQ_INIT
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 636
- SIGQ_MOVE
 - ng_sscop_cust.h, 636
- SIGQ_PREPEND
 - ng_sscop_cust.h, 636
- SIOC_HCI_RAW_NODE_FLUSH_NEIGHBOR_-
CACHE
 - ng_btsocket.h, 798
- SIOC_HCI_RAW_NODE_GET_BDADDR
 - ng_btsocket.h, 798
- SIOC_HCI_RAW_NODE_GET_BUFFER
 - ng_btsocket.h, 798
- SIOC_HCI_RAW_NODE_GET_CON_LIST
 - ng_btsocket.h, 798
- SIOC_HCI_RAW_NODE_GET_DEBUG
 - ng_btsocket.h, 798
- SIOC_HCI_RAW_NODE_GET_FEATURES
 - ng_btsocket.h, 799
- SIOC_HCI_RAW_NODE_GET_LINK_-
POLICY_MASK
 - ng_btsocket.h, 799
- SIOC_HCI_RAW_NODE_GET_NEIGHBOR_-
CACHE
 - ng_btsocket.h, 799
- SIOC_HCI_RAW_NODE_GET_PACKET_MASK
 - ng_btsocket.h, 799
- SIOC_HCI_RAW_NODE_GET_ROLE_SWITCH
 - ng_btsocket.h, 799
- SIOC_HCI_RAW_NODE_GET_STAT
 - ng_btsocket.h, 800

- SIOC_HCI_RAW_NODE_GET_STATE
 - ng_btsocket.h, 800
- SIOC_HCI_RAW_NODE_INIT
 - ng_btsocket.h, 800
- SIOC_HCI_RAW_NODE_LIST_NAMES
 - ng_btsocket.h, 800
- SIOC_HCI_RAW_NODE_RESET_STAT
 - ng_btsocket.h, 800
- SIOC_HCI_RAW_NODE_SET_DEBUG
 - ng_btsocket.h, 800
- SIOC_HCI_RAW_NODE_SET_LINK_POLICY_-
 - MASK
 - ng_btsocket.h, 801
- SIOC_HCI_RAW_NODE_SET_PACKET_MASK
 - ng_btsocket.h, 801
- SIOC_HCI_RAW_NODE_SET_ROLE_SWITCH
 - ng_btsocket.h, 801
- SIOC_L2CAP_L2CA_GET_INFO
 - ng_btsocket.h, 801
- SIOC_L2CAP_L2CA_PING
 - ng_btsocket.h, 801
- SIOC_L2CAP_NODE_GET_AUTO_DISCON_-
 - TIMO
 - ng_btsocket.h, 802
- SIOC_L2CAP_NODE_GET_CHAN_LIST
 - ng_btsocket.h, 802
- SIOC_L2CAP_NODE_GET_CON_LIST
 - ng_btsocket.h, 802
- SIOC_L2CAP_NODE_GET_DEBUG
 - ng_btsocket.h, 802
- SIOC_L2CAP_NODE_GET_FLAGS
 - ng_btsocket.h, 802
- SIOC_L2CAP_NODE_SET_AUTO_DISCON_-
 - TIMO
 - ng_btsocket.h, 803
- SIOC_L2CAP_NODE_SET_DEBUG
 - ng_btsocket.h, 803
- slc
 - ng_vjc_private, 412
- slen
 - ng_async_private, 118
- SLIST_HEAD
 - ng_bridge.c, 1220
- SMALL
 - netflow.c, 1093
- smru
 - ng_async_cfg, 116
- snd_queue
 - privdata, 478
- so
 - ng_btsocket_hci_raw_pcb, 161
 - ng_btsocket_l2cap_pcb, 166
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_btsocket_rfcomm_pcb, 184
 - ng_ksocket_private, 259
 - so2hci_raw_pcb
 - ng_btsocket_hci_raw.h, 806
 - so2l2cap_pcb
 - ng_btsocket_l2cap.h, 814
 - so2l2cap_raw_pcb
 - ng_btsocket_l2cap.h, 815
 - so2rfcomm_pcb
 - ng_btsocket_rfcomm.h, 837
 - SO_HCI_RAW_DIRECTION
 - ng_btsocket.h, 803
 - SO_HCI_RAW_FILTER
 - ng_btsocket.h, 803
 - SO_L2CAP_FLUSH
 - ng_btsocket.h, 803
 - SO_L2CAP_IFLOW
 - ng_btsocket.h, 803
 - SO_L2CAP_IMTU
 - ng_btsocket.h, 804
 - SO_L2CAP_OFLOW
 - ng_btsocket.h, 804
 - SO_L2CAP_OMTU
 - ng_btsocket.h, 804
 - SO_RFCOMM_FC_INFO
 - ng_btsocket.h, 804
 - SO_RFCOMM_MTU
 - ng_btsocket.h, 804
 - sockaddr_hci, 496
 - hci_family, 496
 - hci_len, 496
 - hci_node, 496
 - sockaddr_l2cap, 497
 - l2cap_bdaddr, 497
 - l2cap_family, 497
 - l2cap_len, 497
 - l2cap_psm, 497
 - sockaddr_ng, 499
 - sg_data, 499
 - sg_family, 499
 - sg_len, 499
 - sockaddr_rfcomm, 500
 - rfcomm_bdaddr, 500
 - rfcomm_channel, 500
 - rfcomm_family, 500
 - rfcomm_len, 500
 - sockdata
 - ngpcb, 450
 - SOL_HCI_RAW
 - ng_btsocket.h, 804
 - SOL_L2CAP
 - ng_btsocket.h, 804
 - SOL_RFCOMM
 - ng_btsocket.h, 804
 - SORCVBUF_SIZE

- ng_netflow.h, 1113
- sotongpcb
 - ng_socket.c, 1583
- spare
 - ng_mesg::ng_msghdr, 322
- spare2
 - ng_mesg::ng_msghdr, 322
- src
 - ng_btsocket_l2cap_pcb, 166
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_btsocket_l2cap_rtenry, 178
 - ng_btsocket_rfcomm_pcb, 184
- src_addr
 - netflow_v1_record, 109
 - netflow_v5_record, 115
- src_as
 - netflow_v5_record, 115
- src_mask
 - flow_entry_data, 78
 - netflow_v5_record, 115
- sscf
 - priv, 471
- sscfu_arg, 501
 - data, 501
 - sig, 501
- sscfu_send_lower
 - ng_sscfu.c, 615
- sscfu_send_upper
 - ng_sscfu.c, 615
- sscfu_verbose
 - ng_sscfu.c, 615
- sscfu_window
 - ng_sscfu.c, 615
- sscop
 - priv, 471
- sscop_arg, 502
 - arg, 502
 - data, 502
 - sig, 502
- sscop_marg, 503
 - data, 503
 - sig, 503
- sscop_merr, 504
 - cnt, 504
 - err, 504
 - sig, 504
- sscop_send_lower
 - ng_sscop.c, 626
- sscop_send_manage
 - ng_sscop.c, 626
- sscop_send_upper
 - ng_sscop.c, 626
- sscop_timer_t
 - ng_sscop_cust.h, 638
- sscop_verbose
 - ng_sscop.c, 627
- ssr
 - greheader, 86
- ssth
 - l2tp_seq, 100
- staleness
 - ng_bridge_host, 132
- startkey
 - ng_mppc_config, 324
- startTime
 - ng_pptpgre_private, 372
 - ng_source_stats, 385
- stat
 - bt3c_softc, 57
 - ng_btsocket_hci_raw_node_stat, 158
 - ng_h4_info, 209
 - ng_hci_unit, 238
- state
 - bt3c_softc, 57
 - ng_btsocket_hci_raw_node_state, 159
 - ng_btsocket_l2cap_pcb, 166
 - ng_btsocket_rfcomm_pcb, 185
 - ng_btsocket_rfcomm_session, 187
 - ng_h4_info, 209
 - ng_hci_node_con_ep, 227
 - ng_hci_unit, 238
 - ng_hci_unit_con, 243
 - ng_l2cap_chan, 267
 - ng_l2cap_con, 274
 - ng_l2cap_node_chan_ep, 304
 - ng_l2cap_node_con_ep, 306
 - ng_pppoe.c, 1520
 - ngm_atm_vcc_change, 432
 - sess_con, 493
- stats, 505
 - aa_dropped, 505
 - aa_signals, 505
 - data_delivered, 505
 - errors, 505
 - hookinfo, 87
 - hpriv_p, 89
 - in_dropped, 505
 - in_packets, 505
 - maa_dropped, 506
 - maa_signals, 506
 - ng_async_private, 118
 - ng_bpf_hookinfo, 126
 - ng_bridge_link, 134
 - ng_deflate_private, 195
 - ng_hole_hookinfo, 244
 - ng_l2tp_hook_private, 310
 - ng_l2tp_private, 312
 - ng_one2many_link, 342

- ng_ppp_link, 358
- ng_pptpgre_private, 372
- ng_pred1_private, 377
- ng_vatmpif_hook, 411
- out_dropped, 506
- out_packets, 506
- priv, 471
- privdata, 479
- status
 - __attribute__, 52
 - bt3c_softc, 57
 - ng_hci_lp_con_cfm_ep, 212
 - ng_hci_lp_con_rsp_ep, 215
 - ng_hci_lp_qos_cfm_ep, 218
 - ng_l2cap_l2ca_con_op, 284
 - ng_l2cap_l2ca_con_rsp_ip, 285
 - rfcomm_mcc_ri, 487
- STEPBY
 - ng_lmi.c, 1397
- strip
 - ng_tag_hookin, 393
 - ng_tag_hookinfo, 396
- supertype
 - ng_parse_type, 351
- sync
 - ng_async_private, 119
- SYNC_BUF_SIZE
 - ng_async.c, 1160
- sync_con_queue
 - ng_hci_evt.c, 738
- syncFrames
 - ng_async_stat, 120
- syncOctets
 - ng_async_stat, 121
- syncOverflows
 - ng_async_stat, 121
- SYNPkts
 - ng_tcpmss_hookstat, 400
- sys_uptime
 - netflow_v1_header, 106
 - netflow_v5_header, 111
- SYSCTL_INT
 - ng_base.c, 1200
 - ng_bluetooth.c, 660
 - ng_socket.c, 1589, 1590
- SYSCTL_NODE
 - ng_atm.c, 575
 - ng_base.c, 1200
 - ng_bluetooth.c, 660
 - ng_btsocket.c, 1012, 1013
 - ng_socket.c, 1590
- SYSCTL_PROC
 - ng_bluetooth.c, 660
- T_EOF
 - ng_parse.h, 1479
- T_EQUALS
 - ng_parse.h, 1479
- T_ERROR
 - ng_parse.h, 1479
- T_LBRACE
 - ng_parse.h, 1479
- T_LBRACKET
 - ng_parse.h, 1479
- T_RBRACE
 - ng_parse.h, 1479
- T_RBRACKET
 - ng_parse.h, 1479
- T_STRING
 - ng_parse.h, 1479
- T_WORD
 - ng_parse.h, 1479
- tab
 - ng_bridge_private, 140
- tag
 - ng_tag_prio, 398
 - pppoe_hdr, 466
 - sa_tag, 490
- tag_cookie
 - ng_tag_hookin, 393
 - ng_tag_hookout, 397
- tag_data
 - ng_tag_hookin, 393
 - ng_tag_hookout, 397
 - pppoe_tag, 468
- tag_id
 - ng_tag_hookin, 393
 - ng_tag_hookout, 397
- tag_len
 - ng_tag_hookin, 393
 - ng_tag_hookout, 397
 - pppoe_tag, 468
- tag_type
 - pppoe_tag, 468
- TAGI_HUNIQ
 - ng_pppoe.h, 1536
- TAGI_SVC
 - ng_pppoe.h, 1536
- TAGO_ACNAME
 - ng_pppoe.h, 1536
- TAGO_COOKIE
 - ng_pppoe.h, 1536
- TAGO_HUNIQ
 - ng_pppoe.h, 1536
- TAGO_SVC
 - ng_pppoe.h, 1536
- TAGR_COOKIE
 - ng_pppoe.h, 1536

- TAGR_HUNIQ
 - ng_pppoe.h, 1536
- TAGR_SVC
 - ng_pppoe.h, 1537
- tags
 - sess_neg, 495
- TAGS_ACNAME
 - ng_pppoe.h, 1537
- TAGS_COOKIE
 - ng_pppoe.h, 1537
- TAGS_HUNIQ
 - ng_pppoe.h, 1537
- TAGS_SVC
 - ng_pppoe.h, 1537
- TAGSIZ
 - ng_ipfw.h, 1358
- tail
 - ng_bt_itemq, 145
 - ng_bt_mbufq, 146
- TAILQ_ENTRY
 - flow_entry, 76
 - ng_l2cap_cmd, 270
 - ng_node, 339
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 638
- TAILQ_HEAD
 - flow_hash_entry, 79
 - ng_base.c, 1200
 - ng_l2cap_con, 273
 - ng_sscfu_cust.h, 620
 - ng_sscop_cust.h, 638
- tbe
 - ngm_atm_cpcs_init, 428
- tcp_flags
 - flow_entry_data, 78
- TCPMSS_ADJUST_CHECKSUM
 - ng_tcpmss.c, 1631
- TESTING
 - ng_base.c, 1182
- text_status
 - ng_atm.c, 575
 - ng_sscfu.c, 615
 - ng_sscop.c, 627
 - ng_uni.c, 645
- thisHook
 - ng_bpf_hookprog, 127
 - ng_tag_hookin, 394
 - ng_tag_hookout, 397
- time0
 - cisco_packet, 64
- time1
 - cisco_packet, 64
- timeout
 - __attribute__, 52
 - ng_btsocket_l2cap_raw_auto_discon_timo, 168
 - sess_neg, 495
- timer
 - ng_bridge_private, 140
- TIMER_FUNC
 - ng_sscop_cust.h, 637
- TIMER_FUNC_CALL
 - ng_uni_cust.h, 654
- TIMER_FUNC_PARTY
 - ng_uni_cust.h, 654
- TIMER_FUNC_UNI
 - ng_uni_cust.h, 654
- TIMER_INIT
 - ng_sscop_cust.h, 637
- TIMER_ISACT
 - ng_sscop_cust.h, 637
 - ng_uni_cust.h, 655
- TIMER_RESTART
 - ng_sscop_cust.h, 637
- TIMER_STOP
 - ng_sscop_cust.h, 637
- timeSent
 - ng_pptpgre_ackp, 367
- timestamp
 - ng_ppp_frag, 355
- timo
 - ng_btsocket_l2cap_pcb, 167
 - ng_btsocket_rfcomm_pcb, 185
 - ng_h4_info, 210
 - ng_l2cap_cmd, 270
- tmtu
 - ngm_atm_cpcs_init, 428
- token
 - __attribute__, 52
 - ng_btsocket_hci_raw_pcb, 161
 - ng_btsocket_l2cap_pcb, 167
 - ng_btsocket_l2cap_raw_pcb, 175
 - ng_l2cap_cmd, 270
 - ng_mesg::ng_msghdr, 322
- token_bucket_size
 - __attribute__, 52
- token_rate
 - __attribute__, 52
 - ng_hci_lp_qos_req_ep, 220
- tos
 - flow_rec, 82
 - netflow_v1_record, 109
 - netflow_v5_record, 115
- tp
 - ng_h4_info, 210
 - ngt_sc, 457
- traffic
 - ngm_atm_cpcs_init, 428

- TRAP_ERROR
 - ng_base.c, 1182
- trm
 - ngm_atm_cpcs_init, 428
- TUNABLE_INT
 - ng_base.c, 1200
- tunnel_id
 - ng_l2tp_config, 309
- tx_cred
 - ng_btsocket_rfcomm_fc_info, 180
 - ng_btsocket_rfcomm_pcb, 185
- tx_pkt
 - ng_l2cap_con, 275
- type
 - __attribute__, 52
 - cisco_packet, 64
 - ng_parse_struct_field, 349
 - ngm_mkpeer, 439
 - ngpcb, 450
 - nodeinfo, 461
 - pppoe_hdr, 466
 - rfcomm_mcc_hdr, 483
- type_name
 - typeinfo, 507
- typecookie
 - ng_msg::ng_msghdr, 323
- typeinfo, 507
 - numnodes, 507
 - type_name, 507
 - typelist, 508
- typelist, 508
 - numtypes, 508
 - typeinfo, 508
- typestruct
 - ng_async.c, 1163
 - ng_bpf.c, 1210
 - ng_bt3c_pccard.c, 673
 - ng_btsocket_hci_raw.c, 1028
 - ng_btsocket_l2cap.c, 1049
 - ng_btsocket_l2cap_raw.c, 1061
 - ng_cisco.c, 1236
 - ng_echo.c, 1259
 - ng_eiface.c, 1266
 - ng_etf.c, 1275
 - ng_fec.c, 1296
 - ng_frame_relay.c, 1305
 - ng_h4.c, 686
 - ng_hci_main.c, 749
 - ng_hole.c, 1329
 - ng_iface.c, 1343
 - ng_ip_input.c, 1349
 - ng_l2cap_main.c, 966
 - ng_lmi.c, 1402
 - ng_nat.c, 1434
 - ng_pppoe.c, 1529
 - ng_rfc1490.c, 1569
 - ng_sample.c, 1577
 - ng_socket.c, 1592
 - ng_split.c, 1609
 - ng_sppp.c, 1617
 - ng_tag.c, 1627
 - ng_tty.c, 1652
 - ng_ubt.c, 704
 - ng_UI.c, 1659
- UBT_ACL_RECV
 - ng_ubt_var.h, 707
- UBT_ACL_XMIT
 - ng_ubt_var.h, 707
- UBT_ANY_DEV
 - ng_ubt_var.h, 707
- UBT_BULK_BUFFER_SIZE
 - ng_ubt_var.h, 707
- UBT_BULK_DEV
 - ng_ubt_var.h, 708
- ubt_bulk_in_complete
 - ng_ubt.c, 696
- ubt_bulk_in_complete2
 - ng_ubt.c, 696
- ubt_bulk_in_start
 - ng_ubt.c, 696
- ubt_bulk_out_complete
 - ng_ubt.c, 697
- ubt_bulk_out_complete2
 - ng_ubt.c, 697
- ubt_bulk_out_start
 - ng_ubt.c, 697
- UBT_CMD_XMIT
 - ng_ubt_var.h, 708
- UBT_CTRL_BUFFER_SIZE
 - ng_ubt_var.h, 708
- UBT_CTRL_DEV
 - ng_ubt_var.h, 708
- UBT_DEFAULT_QLEN
 - ng_ubt_var.h, 708
- UBT_EVT_RECV
 - ng_ubt_var.h, 708
- UBT_HAVE_FRAME_TYPE
 - ng_ubt_var.h, 708
- UBT_HCI_REQUEST
 - ng_ubt_var.h, 708
- ubt_intr_complete
 - ng_ubt.c, 698
- ubt_intr_complete2
 - ng_ubt.c, 698
- UBT_INTR_DEV
 - ng_ubt_var.h, 708
- ubt_intr_start

- ng_ubt.c, 698
- UBT_ISOC_BUFFER_SIZE
 - ng_ubt_var.h, 709
- ubt_isoc_in_complete
 - ng_ubt.c, 699
- ubt_isoc_in_complete2
 - ng_ubt.c, 699
- ubt_isoc_in_start
 - ng_ubt.c, 699
- ubt_isoc_out_complete
 - ng_ubt.c, 699
- ubt_isoc_out_complete2
 - ng_ubt.c, 700
- ubt_isoc_out_start
 - ng_ubt.c, 700
- ubt_modevent
 - ng_ubt.c, 700
- UBT_NEED_FRAME_TYPE
 - ng_ubt_var.h, 709
- ubt_request_complete
 - ng_ubt.c, 701
- ubt_request_complete2
 - ng_ubt.c, 701
- ubt_request_start
 - ng_ubt.c, 701
- ubt_reset
 - ng_ubt.c, 702
- UBT_SCO_RECV
 - ng_ubt_var.h, 709
- UBT_SCO_XMIT
 - ng_ubt_var.h, 709
- ubt_softc, 509
 - sc_aclq, 510
 - sc_bulk_in_buffer, 510
 - sc_bulk_in_ep, 510
 - sc_bulk_in_pipe, 510
 - sc_bulk_in_xfer, 510
 - sc_bulk_out_buffer, 511
 - sc_bulk_out_ep, 511
 - sc_bulk_out_pipe, 511
 - sc_bulk_out_xfer, 511
 - sc_cmdq, 511
 - sc_ctrl_buffer, 511
 - sc_ctrl_xfer, 511
 - sc_debug, 511
 - sc_dev, 511
 - sc_flags, 512
 - sc_hook, 512
 - sc_iface0, 512
 - sc_iface1, 512
 - sc_intr_buffer, 512
 - sc_intr_ep, 512
 - sc_intr_pipe, 512
 - sc_intr_xfer, 512
 - sc_isoc_in_buffer, 513
 - sc_isoc_in_ep, 513
 - sc_isoc_in_frlen, 513
 - sc_isoc_in_pipe, 513
 - sc_isoc_in_xfer, 513
 - sc_isoc_nframes, 513
 - sc_isoc_out_buffer, 513
 - sc_isoc_out_ep, 513
 - sc_isoc_out_frlen, 513
 - sc_isoc_out_pipe, 514
 - sc_isoc_out_xfer, 514
 - sc_isoc_size, 514
 - sc_node, 514
 - sc_scoq, 514
 - sc_stat, 514
 - sc_udev, 514
- ubt_softc_p
 - ng_ubt_var.h, 709
- ubt_softc_t
 - ng_ubt_var.h, 709
- ubtbcmfw.c
 - DRIVER_MODULE, 714
 - UBTBCMFW_BSIZE, 712
 - UBTBCMFW_BULK_OUT, 712
 - UBTBCMFW_BULK_OUT_DEV, 712
 - UBTBCMFW_BULK_OUT_EP, 712
 - ubtbcmfw_cdevsw, 716
 - ubtbcmfw_close, 714, 716
 - UBTBCMFW_CONFIG_NO, 713
 - UBTBCMFW_CTRL_DEV, 713
 - UBTBCMFW_ENDPOINT, 713
 - UBTBCMFW_IFACE_IDX, 713
 - UBTBCMFW_INTR_IN, 713
 - UBTBCMFW_INTR_IN_DEV, 713
 - UBTBCMFW_INTR_IN_EP, 713
 - ubtbcmfw_ioctl, 715, 716
 - UBTBCMFW_MINOR, 713
 - ubtbcmfw_open, 715, 716
 - ubtbcmfw_poll, 715, 716
 - ubtbcmfw_read, 715, 716
 - ubtbcmfw_softc_p, 714
 - UBTBCMFW_UNIT, 714
 - ubtbcmfw_write, 715, 717
 - USB_ATTACH, 715
 - USB_DECLARE_DRIVER, 715
 - USB_DETACH, 716
 - USB_MATCH, 716
 - USB_PRODUCT_BROADCOM_-
BCM2033NF, 714
 - USBDEVICE, 714
 - USBDEVNAME, 714
 - USBDEVUNIT, 714
 - UBTBCMFW_BSIZE
 - ubtbcmfw.c, 712

- UBTBCMFW_BULK_OUT
 - ubtbcmfw.c, 712
- UBTBCMFW_BULK_OUT_DEV
 - ubtbcmfw.c, 712
- UBTBCMFW_BULK_OUT_EP
 - ubtbcmfw.c, 712
- ubtbcmfw_cdevsw
 - ubtbcmfw.c, 716
- ubtbcmfw_close
 - ubtbcmfw.c, 714, 716
- UBTBCMFW_CONFIG_NO
 - ubtbcmfw.c, 713
- UBTBCMFW_CTRL_DEV
 - ubtbcmfw.c, 713
- UBTBCMFW_ENDPOINT
 - ubtbcmfw.c, 713
- UBTBCMFW_IFACE_IDX
 - ubtbcmfw.c, 713
- UBTBCMFW_INTR_IN
 - ubtbcmfw.c, 713
- UBTBCMFW_INTR_IN_DEV
 - ubtbcmfw.c, 713
- UBTBCMFW_INTR_IN_EP
 - ubtbcmfw.c, 713
- ubtbcmfw_ioctl
 - ubtbcmfw.c, 715, 716
- UBTBCMFW_MINOR
 - ubtbcmfw.c, 713
- ubtbcmfw_open
 - ubtbcmfw.c, 715, 716
- ubtbcmfw_poll
 - ubtbcmfw.c, 715, 716
- ubtbcmfw_read
 - ubtbcmfw.c, 715, 716
- ubtbcmfw_softc, 516
 - sc_bulk_out_dev, 516
 - sc_bulk_out_pipe, 516
 - sc_ctrl_dev, 516
 - sc_dev, 516
 - sc_dying, 516
 - sc_flags, 516
 - sc_intr_in_dev, 517
 - sc_intr_in_pipe, 517
 - sc_refcnt, 517
 - sc_udev, 517
- ubtbcmfw_softc_p
 - ubtbcmfw.c, 714
- UBTBCMFW_UNIT
 - ubtbcmfw.c, 714
- ubtbcmfw_write
 - ubtbcmfw.c, 715, 717
- uclass
 - __attribute__, 52
 - ng_hci_lp_con_ind_ep, 213
- uma_ctor_flow
 - netflow.c, 1097
- uma_dtor_flow
 - netflow.c, 1097
- uni
 - priv, 471
- UNI_ALLOC
 - ng_uni_cust.h, 655
- uni_arg, 518
 - cookie, 518
 - data, 518
 - sig, 518
- UNI_DEBUG_DEFINE
 - ng_uni.c, 642
- uni_do_status
 - ng_uni.c, 645
- uni_fini
 - ng_uni.c, 645
- UNI_FREE
 - ng_uni_cust.h, 655
- uni_init
 - ng_uni.c, 645
- uni_msg_alloc
 - ngatmbase.c, 607
 - ngatmbase.h, 609
- uni_msg_append
 - ngatmbase.c, 607
- uni_msg_build
 - ngatmbase.c, 607
 - ngatmbase.h, 609
- uni_msg_destroy
 - ngatmbase.c, 607
 - ngatmbase.h, 609
- uni_msg_extend
 - ngatmbase.c, 607
- uni_msg_init
 - ngatmbase.c, 608
- uni_msg_pack_mbuf
 - ngatmbase.c, 608
 - ngatmbase.h, 609
- uni_msg_unpack_mbuf
 - ngatmbase.c, 608
 - ngatmbase.h, 610
- uni_saal_output
 - ng_uni.c, 645
- uni_timer, 519
 - c, 519
- uni_uni_output
 - ng_uni.c, 646
- uni_verbose
 - ng_uni.c, 646
- UNICORE
 - ng_uni_cust.h, 655
- unimem

- ng_uni_cust.h, 656
- UNIMEM_ALL
 - ng_uni_cust.h, 656
- UNIMEM_CALL
 - ng_uni_cust.h, 656
- unimem_debug, 520
 - file, 520
 - lno, 520
- UNIMEM_INS
 - ng_uni_cust.h, 656
- unimem_names
 - ng_uni.c, 650
- UNIMEM_PARTY
 - ng_uni_cust.h, 656
- UNIMEM_SIG
 - ng_uni_cust.h, 656
- unimem_sizes
 - ng_uni_cust.h, 656
- UNIMEM_TYPES
 - ng_uni_cust.h, 655
- uniq, 521
 - bytes, 521
 - pointer, 521
- unit
 - frmrel_softc, 84
 - ng_eiface_private, 198
 - ng_fec_private, 205
 - ng_hci_unit_con, 243
 - ng_iface_private, 250
 - ng_sppp_private, 389
 - ngd_private, 418
- unit_mode
 - __attribute__, 52
- UNITS_BITSPERWORD
 - ng_fec.c, 1291
- unix_nsecs
 - netflow_v1_header, 106
 - netflow_v5_header, 112
- unix_secs
 - netflow_v1_header, 106
 - netflow_v5_header, 112
- unparse
 - ng_parse_type, 351
- unused
 - ng_l2cap_l2ca_con_ind_ip, 282
 - ng_l2cap_l2ca_con_rsp_ip, 285
- up
 - nglmistat, 423
- updated
 - ng_hci_neighbor, 223
- uplink
 - ng_UI_private, 408
- upper
 - ng_pptpgre_private, 372
 - priv, 472
 - private, 476
- USB_ATTACH
 - ng_ubt.c, 702
 - ubtbcmfw.c, 715
- USB_DECLARE_DRIVER
 - ng_ubt.c, 702
 - ubtbcmfw.c, 715
- USB_DETACH
 - ng_ubt.c, 702
 - ubtbcmfw.c, 716
- USB_MATCH
 - ng_ubt.c, 702
 - ubtbcmfw.c, 716
- USB_PRODUCT_BROADCOM_BCM2033NF
 - ubtbcmfw.c, 714
- USBBASEDEVICE
 - ubtbcmfw.c, 714
- USBDEVNAME
 - ubtbcmfw.c, 714
- USBDEVUNIT
 - ubtbcmfw.c, 714
- value
 - ng_ksocket_alias, 257
 - ng_ksocket_socket, 260
- Vatmpif_aal
 - ng_atmpif_var.h, 549
- vatmpif_aal
 - ng_atmpif_var.h, 550
- VATMPIF_AAL_0
 - ng_atmpif_var.h, 550
- VATMPIF_AAL_4
 - ng_atmpif_var.h, 550
- VATMPIF_AAL_5
 - ng_atmpif_var.h, 550
- vatmpif_bearerclass
 - ng_atmpif_harp.c, 542
- VATMPIF_DEBUG_NONE
 - ng_atmpif.h, 590
- VATMPIF_DEBUG_PACKET
 - ng_atmpif.h, 591
- VATMPIF_DEV_NAME
 - ng_atmpif.h, 591
- vatmpif_harp_attach
 - ng_atmpif_harp.c, 542
 - ng_atmpif_var.h, 550
- vatmpif_harp_closevcc
 - ng_atmpif_harp.c, 543
- vatmpif_harp_detach
 - ng_atmpif_harp.c, 543
 - ng_atmpif_var.h, 551
- vatmpif_harp_instvcc
 - ng_atmpif_harp.c, 543

- vatmpif_harp_ioctl
 - ng_atmpif_harp.c, 543
- vatmpif_harp_openvcc
 - ng_atmpif_harp.c, 543
- vatmpif_harp_output
 - ng_atmpif_harp.c, 544
- vatmpif_harp_recv_drain
 - ng_atmpif_harp.c, 544
 - ng_atmpif_var.h, 551
- vatmpif_harp_recv_stack
 - ng_atmpif_harp.c, 544
- vatmpif_header, 522
 - __pad, 522
 - aal, 522
 - cellhdr, 522
 - cookie, 522
 - seq, 522
- VATMPIF_MAX_VCI
 - ng_atmpif.h, 591
- VATMPIF_MAX_VPI
 - ng_atmpif.h, 591
- vatmpif_nif_zone
 - ng_atmpif.c, 539
 - ng_atmpif_var.h, 551
- vatmpif_nunits
 - ng_atmpif_harp.c, 544
- vatmpif_services
 - ng_atmpif_harp.c, 544
- Vatmpif_stats
 - ng_atmpif.h, 591
- vatmpif_stats, 523
 - hva_st_aal5, 523
 - hva_st_atm, 523
 - hva_st_ng, 523
- vatmpif_svaal0
 - ng_atmpif_harp.c, 545
- vatmpif_svaal4
 - ng_atmpif_harp.c, 545
- vatmpif_svaal5
 - ng_atmpif_harp.c, 545
- VATMPIF_TRAF_ABR
 - ng_atmpif_var.h, 550
- VATMPIF_TRAF_CBR
 - ng_atmpif_var.h, 550
- VATMPIF_TRAF_UBR
 - ng_atmpif_var.h, 550
- VATMPIF_TRAF_VBR
 - ng_atmpif_var.h, 550
- Vatmpif_traffic
 - ng_atmpif_var.h, 549
- Vatmpif_traffic_type
 - ng_atmpif_var.h, 549
- vatmpif_traffic_type
 - ng_atmpif_var.h, 550
- Vatmpif_unit
 - ng_atmpif_var.h, 549
- vatmpif_unit, 524
 - conf, 524
 - link, 524
 - node, 524
 - vu_cmn, 524
- Vatmpif_vcc
 - ng_atmpif_var.h, 549
- vatmpif_vcc, 526
 - vv_aal, 526
 - vv_cmn, 526
 - vv_traffic, 526
 - vv_traffic_type, 526
- vatmpif_vcc_zone
 - ng_atmpif.c, 539
 - ng_atmpif_var.h, 551
- VCC_OPEN
 - ng_atm.c, 571
- vci
 - ngm_atm_acr_change, 424
 - ngm_atm_cpcs_init, 428
 - ngm_atm_vcc_change, 432
 - ngvcc, 458
- vci_bits
 - ngm_atm_config, 425
- ver
 - pppoe_hdr, 467
- VERBOSE
 - ng_sscop.c, 623
- vers
 - greheader, 86
- version
 - netflow_v1_header, 106
 - netflow_v5_header, 112
 - ng_mesg::ng_msghdr, 323
 - ng_type, 405
- vjcomp
 - ng_vjc_private, 412
- vjCompHooked
 - ng_ppp_private, 366
- vjip
 - ng_vjc_private, 413
- vjuncomp
 - ng_vjc_private, 413
- vlan
 - ng_vlan_filter, 414
- vpi
 - ngm_atm_acr_change, 424
 - ngm_atm_cpcs_init, 428
 - ngm_atm_vcc_change, 432
 - ngvcc, 459
- vpi_bits
 - ngm_atm_config, 425

- vu_closevcc
 - ng_atmpif_var.h, 547
- vu_cmn
 - vatmpif_unit, 524
- vu_config
 - ng_atmpif_var.h, 547
- vu_cur_pcr
 - ng_atmpif_var.h, 547
- vu_flags
 - ng_atmpif_var.h, 547
- vu_instvcc
 - ng_atmpif_var.h, 547
- vu_ioctl
 - ng_atmpif_var.h, 547
- vu_mtu
 - ng_atmpif_var.h, 548
- vu_nif_zone
 - ng_atmpif_var.h, 548
- vu_open_vcc
 - ng_atmpif_var.h, 548
- vu_openvcc
 - ng_atmpif_var.h, 548
- vu_output
 - ng_atmpif_var.h, 548
- vu_pif
 - ng_atmpif_var.h, 548
- vu_softc
 - ng_atmpif_var.h, 548
- vu_stats
 - ng_atmpif_var.h, 548
- vu_unit
 - ng_atmpif_var.h, 548
- vu_vcc
 - ng_atmpif_var.h, 548
- vu_vcc_zone
 - ng_atmpif_var.h, 548
- vv_aal
 - vatmpif_vcc, 526
- vv_cmn
 - vatmpif_vcc, 526
- vv_connvc
 - ng_atmpif_var.h, 549
- vv_next
 - ng_atmpif_var.h, 549
- vv_state
 - ng_atmpif_var.h, 549
- vv_toku
 - ng_atmpif_var.h, 549
- vv_traffic
 - vatmpif_vcc, 526
- vv_traffic_type
 - vatmpif_vcc, 526
- vv_upper
 - ng_atmpif_var.h, 549
- want
 - bt3c_softc, 57
 - ng_h4_info, 210
- width
 - segment, 491
- winAck
 - ng_pptpgre_ackp, 368
- windowBits
 - ng_deflate_config, 193
- wmax
 - l2tp_seq, 100
- WRITER_ACTIVE
 - ng_base.c, 1182
- x
 - int16_temp, 96
 - int32_temp, 97
 - int64_temp, 98
- xack
 - l2tp_seq, 100
 - ng_l2tp_seq_config, 313
- xack_timer
 - l2tp_seq, 100
- xmit
 - ng_mppc_private, 327
- xmitAck
 - ng_pptpgre_private, 372
- xmitAlg
 - ng_one2many_config, 341
- xmitBroadcasts
 - ng_bridge_link_stats, 137
- xmitDataTooBig
 - ng_l2tp_stats, 317
- xmitDrops
 - ng_l2tp_stats, 317
 - ng_pptpgre_stats, 374
- xmitFrames
 - ng_bpf_hookstat, 128
 - ng_ppp_link_stat, 362
- xmitInvalid
 - ng_l2tp_stats, 317
- xmitLoneAcks
 - ng_pptpgre_stats, 374
- xmitMulticasts
 - ng_bridge_link_stats, 137
- xmitOctets
 - ng_bpf_hookstat, 128
 - ng_bridge_link_stats, 137
 - ng_l2tp_session_stats, 315
 - ng_l2tp_stats, 317
 - ng_one2many_link_stats, 343
 - ng_ppp_link_stat, 362
 - ng_pptpgre_stats, 374
- xmitPackets

- ng_bridge_link_stats, 137
- ng_l2tp_session_stats, 315
- ng_l2tp_stats, 318
- ng_one2many_link_stats, 343
- ng_pptpgre_stats, 374
- xmitRetransmits
 - ng_l2tp_stats, 318
- xmitSeq
 - ng_pptpgre_private, 372
- xmitShortSeq
 - ng_ppp_bund_conf, 353
- xmitTooBig
 - ng_l2tp_stats, 318
 - ng_pptpgre_stats, 374
- xmitWin
 - ng_pptpgre_ackp, 368
- xmitZLBs
 - ng_l2tp_stats, 318
- xoff_char
 - rfcomm_mcc_rpn, 488
- xon_char
 - rfcomm_mcc_rpn, 489
- xseq
 - ng_ppp_mp_state, 363
 - ng_ppp_private, 366
- xwin
 - l2tp_seq, 101
- XXX, 527
 - channel, 527
 - debughook, 527
 - downstream_hook, 528
 - flags, 528
 - node, 528
 - packets_in, 528
 - packets_out, 528
- XXX_hookinfo, 529
 - channel, 529
 - dldi, 529
 - hook, 529
- XXX_NUM_DLCIS
 - ng_sample.h, 1579
- xxx_p
 - ng_sample.c, 1574
- y
 - int16_temp, 96
 - int32_temp, 97
 - int64_temp, 98
- z_alloc
 - ng_deflate.c, 1244
- z_free
 - ng_deflate.c, 1244
- zone