

FreeBSD kernel IPv6 code Reference Manual

Generated by Doxygen 1.4.7

Sat Feb 24 19:41:09 2007

Contents

1	FreeBSD kernel IPv6 code Main Page	1
2	FreeBSD kernel IPv6 code Directory Hierarchy	3
2.1	FreeBSD kernel IPv6 code Directories	3
3	FreeBSD kernel IPv6 code Data Structure Index	5
3.1	FreeBSD kernel IPv6 code Data Structures	5
4	FreeBSD kernel IPv6 code File Index	9
4.1	FreeBSD kernel IPv6 code File List	9
5	FreeBSD kernel IPv6 code Directory Documentation	11
5.1	/usr/src/sys/netinet6/ Directory Reference	11
5.2	/usr/src/ Directory Reference	13
5.3	/usr/src/sys/ Directory Reference	14
5.4	/usr/ Directory Reference	15
6	FreeBSD kernel IPv6 code Data Structure Documentation	17
6.1	addrsel_policyent Struct Reference	17
6.2	aesctr_ctx Struct Reference	18
6.3	aesxcbc_ctx Struct Reference	19
6.4	ah Struct Reference	21
6.5	ah_algorithm Struct Reference	22
6.6	ah_algorithm_state Struct Reference	24
6.7	cblock Union Reference	25
6.8	dadq Struct Reference	26
6.9	esp Struct Reference	27
6.10	esp_algorithm Struct Reference	28
6.11	esptail Struct Reference	31
6.12	icmp6_ifstat Struct Reference	32

6.13	if_set Struct Reference	36
6.14	in6_addr Struct Reference	37
6.15	in6_addrlifetime Struct Reference	38
6.16	in6_addrpolicy Struct Reference	39
6.17	in6_aliasreq Struct Reference	41
6.18	in6_defrouter Struct Reference	43
6.19	in6_drlist Struct Reference	45
6.20	in6_ifaddr Struct Reference	47
6.21	in6_ifextra Struct Reference	50
6.22	in6_ifreq Struct Reference	52
6.23	in6_ifstat Struct Reference	54
6.24	in6_multi Struct Reference	57
6.25	in6_multi_mship Struct Reference	59
6.26	in6_multistep Struct Reference	60
6.27	in6_nbrinfo Struct Reference	61
6.28	in6_ndifreq Struct Reference	63
6.29	in6_ndireq Struct Reference	64
6.30	in6_ondireq Struct Reference	65
6.31	in6_oprlist Struct Reference	67
6.32	in6_pktinfo Struct Reference	69
6.33	in6_prefix Struct Reference	70
6.34	in6_prefixreq Struct Reference	72
6.35	in6_prflags Struct Reference	74
6.36	in6_prflags::prf_ra Struct Reference	76
6.37	in6_prflags::prf_rr Struct Reference	77
6.38	in6_prlist Struct Reference	78
6.39	in6_rrenumreq Struct Reference	80
6.40	in6_rrenumreq::irr_raflagmask Struct Reference	82
6.41	inet6_ndpr_msghdr Struct Reference	83
6.42	inpcbpolicy Struct Reference	85
6.43	ip6_exthdrs Struct Reference	87
6.44	ip6_moptions Struct Reference	88
6.45	ip6_mtuinto Struct Reference	89
6.46	ip6_pktopts Struct Reference	90
6.47	ip6asfrag Struct Reference	93
6.48	ip6aux Struct Reference	95

6.49	ip6ctlparam Struct Reference	97
6.50	ip6po_nhinfo Struct Reference	99
6.51	ip6po_rhinfo Struct Reference	100
6.52	ip6protosw Struct Reference	101
6.53	ip6q Struct Reference	103
6.54	ip6stat Struct Reference	106
6.55	ipcomp Struct Reference	112
6.56	ipcomp_algorithm Struct Reference	113
6.57	ipsec_history Struct Reference	114
6.58	ipsec_output_state Struct Reference	115
6.59	ipsecaux Struct Reference	116
6.60	ipsecrequest Struct Reference	117
6.61	ipsecstat Struct Reference	119
6.62	ipv6_mreq Struct Reference	123
6.63	llinfo_nd6 Struct Reference	124
6.64	mf6c Struct Reference	126
6.65	mf6ctl Struct Reference	129
6.66	mif6 Struct Reference	130
6.67	mif6ctl Struct Reference	132
6.68	mrt6msg Struct Reference	133
6.69	mrt6stat Struct Reference	135
6.70	mtuex_arg Struct Reference	137
6.71	nd_defrouter Struct Reference	138
6.72	nd_ifinfo Struct Reference	140
6.73	nd_opts Union Reference	142
6.74	nd_pfxrouter Struct Reference	144
6.75	nd_prefix Struct Reference	145
6.76	nd_prefixctl Struct Reference	148
6.77	newah Struct Reference	150
6.78	newesp Struct Reference	151
6.79	omrt6msg Struct Reference	152
6.80	pim Struct Reference	154
6.81	pim6stat Struct Reference	155
6.82	randomtab Struct Reference	157
6.83	rip6stat Struct Reference	160
6.84	rtdetq Struct Reference	162

6.85	rtqk_arg Struct Reference	163
6.86	scope6_id Struct Reference	165
6.87	secpolicy Struct Reference	166
6.88	secpolicyindex Struct Reference	169
6.89	secspacq Struct Reference	170
6.90	sioc_mif_req6 Struct Reference	171
6.91	sioc_sg_req6 Struct Reference	172
6.92	sockaddr_in6 Struct Reference	174
6.93	walkarg Struct Reference	176
7	FreeBSD kernel IPv6 code File Documentation	177
7.1	notreviewed.dox File Reference	177
7.2	/usr/src/sys/netinet6/ah.h File Reference	178
7.3	/usr/src/sys/netinet6/ah6.h File Reference	180
7.4	/usr/src/sys/netinet6/ah_aesxcbcmac.c File Reference	181
7.5	/usr/src/sys/netinet6/ah_aesxcbcmac.h File Reference	183
7.6	/usr/src/sys/netinet6/ah_core.c File Reference	184
7.7	/usr/src/sys/netinet6/ah_input.c File Reference	192
7.8	/usr/src/sys/netinet6/ah_output.c File Reference	195
7.9	/usr/src/sys/netinet6/dest6.c File Reference	198
7.10	/usr/src/sys/netinet6/esp.h File Reference	201
7.11	/usr/src/sys/netinet6/esp6.h File Reference	203
7.12	/usr/src/sys/netinet6/esp_aesctr.c File Reference	204
7.13	/usr/src/sys/netinet6/esp_aesctr.h File Reference	207
7.14	/usr/src/sys/netinet6/esp_core.c File Reference	208
7.15	/usr/src/sys/netinet6/esp_input.c File Reference	217
7.16	/usr/src/sys/netinet6/esp_output.c File Reference	220
7.17	/usr/src/sys/netinet6/esp_rijndael.c File Reference	223
7.18	/usr/src/sys/netinet6/esp_rijndael.h File Reference	225
7.19	/usr/src/sys/netinet6/frag6.c File Reference	226
7.20	/usr/src/sys/netinet6/icmp6.c File Reference	233
7.21	/usr/src/sys/netinet6/icmp6.h File Reference	245
7.22	/usr/src/sys/netinet6/in6.c File Reference	246
7.23	/usr/src/sys/netinet6/in6.h File Reference	263
7.24	/usr/src/sys/netinet6/in6_cksum.c File Reference	282
7.25	/usr/src/sys/netinet6/in6_gif.c File Reference	284
7.26	/usr/src/sys/netinet6/in6_gif.h File Reference	289

7.27	/usr/src/sys/netinet6/in6_ifattach.c File Reference	290
7.28	/usr/src/sys/netinet6/in6_ifattach.h File Reference	299
7.29	/usr/src/sys/netinet6/in6_pcb.c File Reference	300
7.30	/usr/src/sys/netinet6/in6_pcb.h File Reference	308
7.31	/usr/src/sys/netinet6/in6_proto.c File Reference	310
7.32	/usr/src/sys/netinet6/in6_rmx.c File Reference	324
7.33	/usr/src/sys/netinet6/in6_src.c File Reference	330
7.34	/usr/src/sys/netinet6/in6_var.h File Reference	340
7.35	/usr/src/sys/netinet6/ip6.h File Reference	358
7.36	/usr/src/sys/netinet6/ip6_ecn.h File Reference	359
7.37	/usr/src/sys/netinet6/ip6_forward.c File Reference	360
7.38	/usr/src/sys/netinet6/ip6_id.c File Reference	363
7.39	/usr/src/sys/netinet6/ip6_input.c File Reference	367
7.40	/usr/src/sys/netinet6/ip6_mroute.c File Reference	378
7.41	/usr/src/sys/netinet6/ip6_mroute.h File Reference	392
7.42	/usr/src/sys/netinet6/ip6_output.c File Reference	397
7.43	/usr/src/sys/netinet6/ip6_var.h File Reference	411
7.44	/usr/src/sys/netinet6/ip6protosw.h File Reference	423
7.45	/usr/src/sys/netinet6/ipcomp.h File Reference	424
7.46	/usr/src/sys/netinet6/ipcomp6.h File Reference	426
7.47	/usr/src/sys/netinet6/ipcomp_core.c File Reference	427
7.48	/usr/src/sys/netinet6/ipcomp_input.c File Reference	432
7.49	/usr/src/sys/netinet6/ipcomp_output.c File Reference	435
7.50	/usr/src/sys/netinet6/ipsec.c File Reference	438
7.51	/usr/src/sys/netinet6/ipsec.h File Reference	456
7.52	/usr/src/sys/netinet6/ipsec6.h File Reference	465
7.53	/usr/src/sys/netinet6/mld6.c File Reference	468
7.54	/usr/src/sys/netinet6/mld6_var.h File Reference	475
7.55	/usr/src/sys/netinet6/nd6.c File Reference	478
7.56	/usr/src/sys/netinet6/nd6.h File Reference	495
7.57	/usr/src/sys/netinet6/nd6_nbr.c File Reference	509
7.58	/usr/src/sys/netinet6/nd6_rtr.c File Reference	519
7.59	/usr/src/sys/netinet6/pim6.h File Reference	535
7.60	/usr/src/sys/netinet6/pim6_var.h File Reference	536
7.61	/usr/src/sys/netinet6/raw_ip6.c File Reference	537
7.62	/usr/src/sys/netinet6/raw_ip6.h File Reference	545

7.63	/usr/src/sys/netinet6/route6.c File Reference	546
7.64	/usr/src/sys/netinet6/scope6.c File Reference	549
7.65	/usr/src/sys/netinet6/scope6_var.h File Reference	555
7.66	/usr/src/sys/netinet6/sctp6_usrreq.c File Reference	557
7.67	/usr/src/sys/netinet6/sctp6_var.h File Reference	563
7.68	/usr/src/sys/netinet6/tcp6_var.h File Reference	564
7.69	/usr/src/sys/netinet6/udp6_output.c File Reference	565
7.70	/usr/src/sys/netinet6/udp6_usrreq.c File Reference	569
7.71	/usr/src/sys/netinet6/udp6_var.h File Reference	577

Chapter 1

FreeBSD kernel IPv6 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 IPv6 code Directory Hierarchy

2.1 FreeBSD kernel IPv6 code Directories

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

usr	15
src	13
sys	14
netinet6	11

Chapter 3

FreeBSD kernel IPv6 code Data Structure Index

3.1 FreeBSD kernel IPv6 code Data Structures

Here are the data structures with brief descriptions:

addrsel_policyent	17
aesctr_ctx	18
aesxcbc_ctx	19
ah	21
ah_algorithm	22
ah_algorithm_state	24
cblock	25
dadq	26
esp	27
esp_algorithm	28
esptail	31
icmp6_ifstat	32
if_set	36
in6_addr	37
in6_addrlifetime	38
in6_addrpolicy	39
in6_aliasreq	41
in6_defrouter	43
in6_drlist	45
in6_ifaddr	47
in6_ifextra	50
in6_ifreq	52
in6_ifstat	54
in6_multi	57
in6_multi_mship	59
in6_multistep	60
in6_nbrinfo	61
in6_ndifreq	63
in6_ndireq	64
in6_ondireq	65
in6_oprlist	67

in6_pktinfo	69
in6_prefix	70
in6_prefixreq	72
in6_prflags	74
in6_prflags::prf_ra	76
in6_prflags::prf_rr	77
in6_prlist	78
in6_rrenumreq	80
in6_rrenumreq::irr_raflagmask	82
inet6_ndpr_msghdr	83
inpcbpolicy	85
ip6_exthdrs	87
ip6_moptions	88
ip6_mtuinfo	89
ip6_pktopts	90
ip6asfrag	93
ip6aux	95
ip6ctlparam	97
ip6po_nhinfo	99
ip6po_rhinfo	100
ip6protosw	101
ip6q	103
ip6stat	106
ipcomp	112
ipcomp_algorithm	113
ipsec_history	114
ipsec_output_state	115
ipsecaux	116
ipsecrequest	117
ipsecstat	119
ipv6_mreq	123
llinfo_nd6	124
mf6c	126
mf6ctl	129
mif6	130
mif6ctl	132
mrt6msg	133
mrt6stat	135
mtuex_arg	137
nd_defrouter	138
nd_ifinfo	140
nd_opts	142
nd_pfxrouter	144
nd_prefix	145
nd_prefixctl	148
newah	150
newesp	151
omrt6msg	152
pim	154
pim6stat	155
randomtab	157
rip6stat	160
rtdetq	162
rtqk_arg	163

scope6_id	165
secpolicy	166
secpolicyindex	169
secspacq	170
sioc_mif_req6	171
sioc_sg_req6	172
sockaddr_in6	174
walkarg	176

Chapter 4

FreeBSD kernel IPv6 code File Index

4.1 FreeBSD kernel IPv6 code File List

Here is a list of all files with brief descriptions:

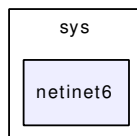
/usr/src/sys/netinet6/ah.h	178
/usr/src/sys/netinet6/ah6.h	180
/usr/src/sys/netinet6/ah_aesxcbcmac.c	181
/usr/src/sys/netinet6/ah_aesxcbcmac.h	183
/usr/src/sys/netinet6/ah_core.c	184
/usr/src/sys/netinet6/ah_input.c	192
/usr/src/sys/netinet6/ah_output.c	195
/usr/src/sys/netinet6/dest6.c	198
/usr/src/sys/netinet6/esp.h	201
/usr/src/sys/netinet6/esp6.h	203
/usr/src/sys/netinet6/esp_aesctr.c	204
/usr/src/sys/netinet6/esp_aesctr.h	207
/usr/src/sys/netinet6/esp_core.c	208
/usr/src/sys/netinet6/esp_input.c	217
/usr/src/sys/netinet6/esp_output.c	220
/usr/src/sys/netinet6/esp_rijndael.c	223
/usr/src/sys/netinet6/esp_rijndael.h	225
/usr/src/sys/netinet6/frag6.c	226
/usr/src/sys/netinet6/icmp6.c	233
/usr/src/sys/netinet6/icmp6.h	245
/usr/src/sys/netinet6/in6.c	246
/usr/src/sys/netinet6/in6.h	263
/usr/src/sys/netinet6/in6_cksum.c	282
/usr/src/sys/netinet6/in6_gif.c	284
/usr/src/sys/netinet6/in6_gif.h	289
/usr/src/sys/netinet6/in6_ifattach.c	290
/usr/src/sys/netinet6/in6_ifattach.h	299
/usr/src/sys/netinet6/in6_pcb.c	300
/usr/src/sys/netinet6/in6_pcb.h	308
/usr/src/sys/netinet6/in6_proto.c	310
/usr/src/sys/netinet6/in6_rmx.c	324
/usr/src/sys/netinet6/in6_src.c	330
/usr/src/sys/netinet6/in6_var.h	340

/usr/src/sys/netinet6/ip6.h	358
/usr/src/sys/netinet6/ip6_ecn.h	359
/usr/src/sys/netinet6/ip6_forward.c	360
/usr/src/sys/netinet6/ip6_id.c	363
/usr/src/sys/netinet6/ip6_input.c	367
/usr/src/sys/netinet6/ip6_mroute.c	378
/usr/src/sys/netinet6/ip6_mroute.h	392
/usr/src/sys/netinet6/ip6_output.c	397
/usr/src/sys/netinet6/ip6_var.h	411
/usr/src/sys/netinet6/ip6protosw.h	423
/usr/src/sys/netinet6/ipcomp.h	424
/usr/src/sys/netinet6/ipcomp6.h	426
/usr/src/sys/netinet6/ipcomp_core.c	427
/usr/src/sys/netinet6/ipcomp_input.c	432
/usr/src/sys/netinet6/ipcomp_output.c	435
/usr/src/sys/netinet6/ipsec.c	438
/usr/src/sys/netinet6/ipsec.h	456
/usr/src/sys/netinet6/ipsec6.h	465
/usr/src/sys/netinet6/mld6.c	468
/usr/src/sys/netinet6/mld6_var.h	475
/usr/src/sys/netinet6/nd6.c	478
/usr/src/sys/netinet6/nd6.h	495
/usr/src/sys/netinet6/nd6_nbr.c	509
/usr/src/sys/netinet6/nd6_rtr.c	519
/usr/src/sys/netinet6/pim6.h	535
/usr/src/sys/netinet6/pim6_var.h	536
/usr/src/sys/netinet6/raw_ip6.c	537
/usr/src/sys/netinet6/raw_ip6.h	545
/usr/src/sys/netinet6/route6.c	546
/usr/src/sys/netinet6/scope6.c	549
/usr/src/sys/netinet6/scope6_var.h	555
/usr/src/sys/netinet6/sctp6_usrreq.c	557
/usr/src/sys/netinet6/sctp6_var.h	563
/usr/src/sys/netinet6/tcp6_var.h	564
/usr/src/sys/netinet6/udp6_output.c	565
/usr/src/sys/netinet6/udp6_usrreq.c	569
/usr/src/sys/netinet6/udp6_var.h	577

Chapter 5

FreeBSD kernel IPv6 code Directory Documentation

5.1 /usr/src/sys/netinet6/ Directory Reference

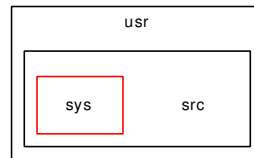


Files

- file [ah.h](#)
- file [ah6.h](#)
- file [ah_aesxcbcmac.c](#)
- file [ah_aesxcbcmac.h](#)
- file [ah_core.c](#)
- file [ah_input.c](#)
- file [ah_output.c](#)
- file [dest6.c](#)
- file [esp.h](#)
- file [esp6.h](#)
- file [esp_aesctr.c](#)
- file [esp_aesctr.h](#)
- file [esp_core.c](#)
- file [esp_input.c](#)
- file [esp_output.c](#)
- file [esp_rijndael.c](#)
- file [esp_rijndael.h](#)
- file [frag6.c](#)
- file [icmp6.c](#)
- file [icmp6.h](#)

- file [in6.c](#)
- file [in6.h](#)
- file [in6_cksum.c](#)
- file [in6_gif.c](#)
- file [in6_gif.h](#)
- file [in6_ifattach.c](#)
- file [in6_ifattach.h](#)
- file [in6_pcb.c](#)
- file [in6_pcb.h](#)
- file [in6_proto.c](#)
- file [in6_rmx.c](#)
- file [in6_src.c](#)
- file [in6_var.h](#)
- file [ip6.h](#)
- file [ip6_ecn.h](#)
- file [ip6_forward.c](#)
- file [ip6_id.c](#)
- file [ip6_input.c](#)
- file [ip6_mroute.c](#)
- file [ip6_mroute.h](#)
- file [ip6_output.c](#)
- file [ip6_var.h](#)
- file [ip6protosw.h](#)
- file [ipcomp.h](#)
- file [ipcomp6.h](#)
- file [ipcomp_core.c](#)
- file [ipcomp_input.c](#)
- file [ipcomp_output.c](#)
- file [ipsec.c](#)
- file [ipsec.h](#)
- file [ipsec6.h](#)
- file [mld6.c](#)
- file [mld6_var.h](#)
- file [nd6.c](#)
- file [nd6.h](#)
- file [nd6_nbr.c](#)
- file [nd6_rtr.c](#)
- file [pim6.h](#)
- file [pim6_var.h](#)
- file [raw_ip6.c](#)
- file [raw_ip6.h](#)
- file [route6.c](#)
- file [scope6.c](#)
- file [scope6_var.h](#)
- file [sctp6_usrreq.c](#)
- file [sctp6_var.h](#)
- file [tcp6_var.h](#)
- file [udp6_output.c](#)
- file [udp6_usrreq.c](#)
- file [udp6_var.h](#)

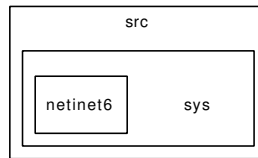
5.2 /usr/src/ Directory Reference



Directories

- directory [sys](#)

5.3 /usr/src/sys/ Directory Reference



Directories

- directory [netinet6](#)

5.4 /usr/ Directory Reference



Directories

- directory [src](#)

Chapter 6

FreeBSD kernel IPv6 code Data Structure Documentation

6.1 addrsel_policyent Struct Reference

6.1.1 Detailed Description

Definition at line 948 of file in6_src.c.

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

- [/usr/src/sys/netinet6/in6_src.c](#)

6.2 aesctr_ctx Struct Reference

Data Fields

- `u_int32_t r_ek` [(RIJNDAEL_MAXNR+1)*4]
- `int r_nr`

6.2.1 Detailed Description

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

6.2.2 Field Documentation

6.2.2.1 `u_int32_t aesctr_ctx::r_ek`[(RIJNDAEL_MAXNR+1)*4]

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

Referenced by `esp_aesctr_decrypt()`, `esp_aesctr_encrypt()`, and `esp_aesctr_schedule()`.

6.2.2.2 `int aesctr_ctx::r_nr`

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

Referenced by `esp_aesctr_decrypt()`, `esp_aesctr_encrypt()`, and `esp_aesctr_schedule()`.

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

- `/usr/src/sys/netinet6/esp_aesctr.c`

6.3 aesxcbc_ctx Struct Reference

Data Fields

- `u_int8_t e` [AES_BLOCKSIZE]
- `u_int8_t buf` [AES_BLOCKSIZE]
- `size_t buflen`
- `u_int32_t r_k1s` [(RIJNDAEL_MAXNR+1)*4]
- `u_int32_t r_k2s` [(RIJNDAEL_MAXNR+1)*4]
- `u_int32_t r_k3s` [(RIJNDAEL_MAXNR+1)*4]
- `int r_nr`
- `u_int8_t k2` [AES_BLOCKSIZE]
- `u_int8_t k3` [AES_BLOCKSIZE]

6.3.1 Detailed Description

Definition at line 57 of file ah_aesxcbcmac.c.

6.3.2 Field Documentation

6.3.2.1 `u_int8_t aesxcbc_ctx::buf`[AES_BLOCKSIZE]

Definition at line 59 of file ah_aesxcbcmac.c.

Referenced by ah_aes_xcbc_mac_loop(), and ah_aes_xcbc_mac_result().

6.3.2.2 `size_t aesxcbc_ctx::buflen`

Definition at line 60 of file ah_aesxcbcmac.c.

Referenced by ah_aes_xcbc_mac_loop(), and ah_aes_xcbc_mac_result().

6.3.2.3 `u_int8_t aesxcbc_ctx::e`[AES_BLOCKSIZE]

Definition at line 58 of file ah_aesxcbcmac.c.

Referenced by ah_aes_xcbc_mac_loop(), and ah_aes_xcbc_mac_result().

6.3.2.4 `u_int8_t aesxcbc_ctx::k2`[AES_BLOCKSIZE]

Definition at line 65 of file ah_aesxcbcmac.c.

Referenced by ah_aes_xcbc_mac_result().

6.3.2.5 `u_int8_t aesxcbc_ctx::k3`[AES_BLOCKSIZE]

Definition at line 66 of file ah_aesxcbcmac.c.

Referenced by ah_aes_xcbc_mac_result().

6.3.2.6 `u_int32_t aesxcbc_ctx::r_k1s[(RIJNDAEL_MAXNR+1)*4]`

Definition at line 61 of file `ah_aesxcbcmac.c`.

Referenced by `ah_aes_xcbc_mac_loop()`, and `ah_aes_xcbc_mac_result()`.

6.3.2.7 `u_int32_t aesxcbc_ctx::r_k2s[(RIJNDAEL_MAXNR+1)*4]`

Definition at line 62 of file `ah_aesxcbcmac.c`.

6.3.2.8 `u_int32_t aesxcbc_ctx::r_k3s[(RIJNDAEL_MAXNR+1)*4]`

Definition at line 63 of file `ah_aesxcbcmac.c`.

6.3.2.9 `int aesxcbc_ctx::r_nr`

Definition at line 64 of file `ah_aesxcbcmac.c`.

Referenced by `ah_aes_xcbc_mac_loop()`, and `ah_aes_xcbc_mac_result()`.

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

- `/usr/src/sys/netinet6/ah_aesxcbcmac.c`

6.4 ah Struct Reference

```
#include <ah.h>
```

Data Fields

- [u_int8_t ah_nxt](#)
- [u_int8_t ah_len](#)
- [u_int16_t ah_reserve](#)
- [u_int32_t ah_spi](#)

6.4.1 Detailed Description

Definition at line 44 of file ah.h.

6.4.2 Field Documentation

6.4.2.1 [u_int8_t ah::ah_len](#)

Definition at line 46 of file ah.h.

6.4.2.2 [u_int8_t ah::ah_nxt](#)

Definition at line 45 of file ah.h.

6.4.2.3 [u_int16_t ah::ah_reserve](#)

Definition at line 47 of file ah.h.

6.4.2.4 [u_int32_t ah::ah_spi](#)

Definition at line 48 of file ah.h.

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

- [/usr/src/sys/netinet6/ah.h](#)

6.5 ah_algorithm Struct Reference

```
#include <ah.h>
```

Public Member Functions

- int sumsiz [__P](#) ((struct secasvar *)
- int mature [__P](#) ((struct secasvar *)
- int init [__P](#) ((struct [ah_algorithm_state](#) *, struct secasvar *)
- void update [__P](#) ((struct [ah_algorithm_state](#) *, u_int8_t *, [size_t](#))
- void result [__P](#) ((struct [ah_algorithm_state](#) *, u_int8_t *, [size_t](#))

Data Fields

- int [keymin](#)
- int [keymax](#)
- const char * [name](#)

6.5.1 Detailed Description

Definition at line 69 of file ah.h.

6.5.2 Member Function Documentation

6.5.2.1 void result [ah_algorithm::__P](#) ((struct [ah_algorithm_state](#) *, u_int8_t *, [size_t](#))

6.5.2.2 void update [ah_algorithm::__P](#) ((struct [ah_algorithm_state](#) *, u_int8_t *, [size_t](#))

6.5.2.3 int init [ah_algorithm::__P](#) ((struct [ah_algorithm_state](#) *, struct secasvar *)

6.5.2.4 int mature [ah_algorithm::__P](#) ((struct secasvar *)

6.5.2.5 int sumsiz [ah_algorithm::__P](#) ((struct secasvar *)

6.5.3 Field Documentation

6.5.3.1 int [ah_algorithm::keymax](#)

Definition at line 73 of file ah.h.

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

6.5.3.2 int [ah_algorithm::keymin](#)

Definition at line 72 of file ah.h.

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

6.5.3.3 const char* [ah_algorithm::name](#)

Definition at line 74 of file ah.h.

Referenced by ah_common_mature().

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

- [/usr/src/sys/netinet6/ah.h](#)

6.6 ah_algorithm_state Struct Reference

```
#include <ah.h>
```

Data Fields

- secasvar * [sav](#)
- void * [foo](#)

6.6.1 Detailed Description

Definition at line 64 of file ah.h.

6.6.2 Field Documentation

6.6.2.1 void* [ah_algorithm_state::foo](#)

Definition at line 66 of file ah.h.

6.6.2.2 struct secasvar* [ah_algorithm_state::sav](#)

Definition at line 65 of file ah.h.

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

- [/usr/src/sys/netinet6/ah.h](#)

6.7 cblock Union Reference

Data Fields

- struct {
 - u_int8_t [nonce](#) [4]
 - u_int8_t [iv](#) [8]
 - u_int32_t [ctr](#)
- [v](#)
- u_int8_t [cblock](#) [16]

6.7.1 Detailed Description

Definition at line 57 of file `esp_aesctr.c`.

6.7.2 Field Documentation

6.7.2.1 u_int8_t [cblock::cblock](#)[16]

Definition at line 63 of file `esp_aesctr.c`.

Referenced by `esp_aesctr_decrypt()`, and `esp_aesctr_encrypt()`.

6.7.2.2 u_int32_t [cblock::ctr](#)

Definition at line 61 of file `esp_aesctr.c`.

Referenced by `esp_aesctr_decrypt()`, and `esp_aesctr_encrypt()`.

6.7.2.3 u_int8_t [cblock::iv](#)[8]

Definition at line 60 of file `esp_aesctr.c`.

Referenced by `esp_aesctr_decrypt()`, and `esp_aesctr_encrypt()`.

6.7.2.4 u_int8_t [cblock::nonce](#)[4]

Definition at line 59 of file `esp_aesctr.c`.

Referenced by `esp_aesctr_decrypt()`, and `esp_aesctr_encrypt()`.

6.7.2.5 struct { ... } [cblock::v](#)

Referenced by `esp_aesctr_decrypt()`, and `esp_aesctr_encrypt()`.

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

- `/usr/src/sys/netinet6/esp_aesctr.c`

6.8 dadq Struct Reference

6.8.1 Detailed Description

Definition at line 1055 of file `nd6_nbr.c`.

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

- [/usr/src/sys/netinet6/nd6_nbr.c](#)

6.9 esp Struct Reference

```
#include <esp.h>
```

Data Fields

- [u_int32_t esp_spi](#)

6.9.1 Detailed Description

Definition at line 44 of file esp.h.

6.9.2 Field Documentation

6.9.2.1 [u_int32_t esp::esp_spi](#)

Definition at line 45 of file esp.h.

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

- [/usr/src/sys/netinet6/esp.h](#)

6.10 esp_algorithm Struct Reference

```
#include <esp.h>
```

Public Member Functions

- int mature [__P](#) ((struct secasvar *))
- [size_t](#) schedlen [__P](#) ((const struct [esp_algorithm](#) *))
- int ivlen [__P](#) ((const struct [esp_algorithm](#) *, struct secasvar *))
- int decrypt [__P](#) ((struct mbuf *, [size_t](#), struct secasvar *, const struct [esp_algorithm](#) *, int))
- int encrypt [__P](#) ((struct mbuf *, [size_t](#), [size_t](#), struct secasvar *, const struct [esp_algorithm](#) *, int))
- int schedule [__P](#) ((const struct [esp_algorithm](#) *, struct secasvar *))
- int blockdecrypt [__P](#) ((const struct [esp_algorithm](#) *, struct secasvar *, [u_int8_t](#) *, [u_int8_t](#) *))
- int blockencrypt [__P](#) ((const struct [esp_algorithm](#) *, struct secasvar *, [u_int8_t](#) *, [u_int8_t](#) *))

Data Fields

- [size_t](#) padbound
- int ivlenval
- int keymin
- int keymax
- const char * name

6.10.1 Detailed Description

Definition at line 75 of file esp.h.

6.10.2 Member Function Documentation

6.10.2.1 `int blockencrypt esp_algorithm::__P ((const struct esp_algorithm *, struct secasvar *, u_int8_t *, u_int8_t *))`

6.10.2.2 `int blockdecrypt esp_algorithm::__P ((const struct esp_algorithm *, struct secasvar *, u_int8_t *, u_int8_t *))`

6.10.2.3 `int schedule esp_algorithm::__P ((const struct esp_algorithm *, struct secasvar *))`

6.10.2.4 `int encrypt esp_algorithm::__P ((struct mbuf *, size_t, size_t, struct secasvar *, const struct esp_algorithm *, int))`

6.10.2.5 `int decrypt esp_algorithm::__P ((struct mbuf *, size_t, struct secasvar *, const struct esp_algorithm *, int))`

6.10.2.6 `int ivlen esp_algorithm::__P ((const struct esp_algorithm *, struct secasvar *))`

6.10.2.7 `size_t schedlen esp_algorithm::__P ((const struct esp_algorithm *))`

6.10.2.8 `int mature esp_algorithm::__P ((struct secasvar *))`

6.10.3 Field Documentation

6.10.3.1 `int esp_algorithm::ivlenval`

Definition at line 77 of file esp.h.

Referenced by esp_max_ivlen().

6.10.3.2 `int esp_algorithm::keymax`

Definition at line 80 of file esp.h.

Referenced by esp_aesctr_mature(), esp_cbc_mature(), and esp_descbc_mature().

6.10.3.3 `int esp_algorithm::keymin`

Definition at line 79 of file esp.h.

Referenced by esp_aesctr_mature(), esp_cbc_mature(), and esp_descbc_mature().

6.10.3.4 `const char* esp_algorithm::name`

Definition at line 82 of file esp.h.

Referenced by esp_aesctr_decrypt(), esp_aesctr_encrypt(), esp_aesctr_mature(), esp_cbc_decrypt(), esp_cbc_encrypt(), and esp_cbc_mature().

6.10.3.5 `size_t esp_algorithm::padbound`

Definition at line 76 of file esp.h.

Referenced by `esp_aesctr_decrypt()`, `esp_aesctr_encrypt()`, `esp_cbc_decrypt()`, and `esp_cbc_encrypt()`.

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

- [/usr/src/sys/netinet6/esp.h](#)

6.11 esptail Struct Reference

```
#include <esp.h>
```

Data Fields

- [u_int8_t esp_padlen](#)
- [u_int8_t esp_nxt](#)

6.11.1 Detailed Description

Definition at line 66 of file esp.h.

6.11.2 Field Documentation

6.11.2.1 [u_int8_t esptail::esp_nxt](#)

Definition at line 68 of file esp.h.

6.11.2.2 [u_int8_t esptail::esp_padlen](#)

Definition at line 67 of file esp.h.

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

- [/usr/src/sys/netinet6/esp.h](#)

6.12 icmp6_ifstat Struct Reference

```
#include <in6_var.h>
```

Data Fields

- `u_quad_t ifs6_in_msg`
- `u_quad_t ifs6_in_error`
- `u_quad_t ifs6_in_dstunreach`
- `u_quad_t ifs6_in_adminprohib`
- `u_quad_t ifs6_in_timeexceed`
- `u_quad_t ifs6_in_paramprob`
- `u_quad_t ifs6_in_pkttoobig`
- `u_quad_t ifs6_in_echo`
- `u_quad_t ifs6_in_echoreply`
- `u_quad_t ifs6_in_routersolicit`
- `u_quad_t ifs6_in_routeradvert`
- `u_quad_t ifs6_in_neighborsolicit`
- `u_quad_t ifs6_in_neighboradvert`
- `u_quad_t ifs6_in_redirect`
- `u_quad_t ifs6_in_mldquery`
- `u_quad_t ifs6_in_mldreport`
- `u_quad_t ifs6_in_mlddone`
- `u_quad_t ifs6_out_msg`
- `u_quad_t ifs6_out_error`
- `u_quad_t ifs6_out_dstunreach`
- `u_quad_t ifs6_out_adminprohib`
- `u_quad_t ifs6_out_timeexceed`
- `u_quad_t ifs6_out_paramprob`
- `u_quad_t ifs6_out_pkttoobig`
- `u_quad_t ifs6_out_echo`
- `u_quad_t ifs6_out_echoreply`
- `u_quad_t ifs6_out_routersolicit`
- `u_quad_t ifs6_out_routeradvert`
- `u_quad_t ifs6_out_neighborsolicit`
- `u_quad_t ifs6_out_neighboradvert`
- `u_quad_t ifs6_out_redirect`
- `u_quad_t ifs6_out_mldquery`
- `u_quad_t ifs6_out_mldreport`
- `u_quad_t ifs6_out_mlddone`

6.12.1 Detailed Description

Definition at line 170 of file `in6_var.h`.

6.12.2 Field Documentation

6.12.2.1 `u_quad_t icmp6_ifstat::ifs6_in_adminprohib`

Definition at line 181 of file `in6_var.h`.

6.12.2.2 u_quad_t icmp6_ifstat::ifs6_in_dstunreach

Definition at line 179 of file in6_var.h.

6.12.2.3 u_quad_t icmp6_ifstat::ifs6_in_echo

Definition at line 189 of file in6_var.h.

6.12.2.4 u_quad_t icmp6_ifstat::ifs6_in_echoreply

Definition at line 191 of file in6_var.h.

6.12.2.5 u_quad_t icmp6_ifstat::ifs6_in_error

Definition at line 177 of file in6_var.h.

6.12.2.6 u_quad_t icmp6_ifstat::ifs6_in_mlddone

Definition at line 207 of file in6_var.h.

6.12.2.7 u_quad_t icmp6_ifstat::ifs6_in_mldquery

Definition at line 203 of file in6_var.h.

6.12.2.8 u_quad_t icmp6_ifstat::ifs6_in_mldreport

Definition at line 205 of file in6_var.h.

6.12.2.9 u_quad_t icmp6_ifstat::ifs6_in_msg

Definition at line 175 of file in6_var.h.

6.12.2.10 u_quad_t icmp6_ifstat::ifs6_in_neighboradvert

Definition at line 199 of file in6_var.h.

6.12.2.11 u_quad_t icmp6_ifstat::ifs6_in_neighborsolicit

Definition at line 197 of file in6_var.h.

6.12.2.12 u_quad_t icmp6_ifstat::ifs6_in_paramprob

Definition at line 185 of file in6_var.h.

6.12.2.13 `u_quad_t icmp6_ifstat::ifs6_in_pkttoobig`

Definition at line 187 of file in6_var.h.

6.12.2.14 `u_quad_t icmp6_ifstat::ifs6_in_redirect`

Definition at line 201 of file in6_var.h.

6.12.2.15 `u_quad_t icmp6_ifstat::ifs6_in_routeradvert`

Definition at line 195 of file in6_var.h.

6.12.2.16 `u_quad_t icmp6_ifstat::ifs6_in_routersolicit`

Definition at line 193 of file in6_var.h.

6.12.2.17 `u_quad_t icmp6_ifstat::ifs6_in_timeexceed`

Definition at line 183 of file in6_var.h.

6.12.2.18 `u_quad_t icmp6_ifstat::ifs6_out_adminprohib`

Definition at line 219 of file in6_var.h.

6.12.2.19 `u_quad_t icmp6_ifstat::ifs6_out_dstunreach`

Definition at line 217 of file in6_var.h.

6.12.2.20 `u_quad_t icmp6_ifstat::ifs6_out_echo`

Definition at line 227 of file in6_var.h.

6.12.2.21 `u_quad_t icmp6_ifstat::ifs6_out_echoreply`

Definition at line 229 of file in6_var.h.

6.12.2.22 `u_quad_t icmp6_ifstat::ifs6_out_error`

Definition at line 215 of file in6_var.h.

6.12.2.23 `u_quad_t icmp6_ifstat::ifs6_out_mlddone`

Definition at line 245 of file in6_var.h.

6.12.2.24 u_quad_t icmp6_ifstat::ifs6_out_mldquery

Definition at line 241 of file in6_var.h.

6.12.2.25 u_quad_t icmp6_ifstat::ifs6_out_mldreport

Definition at line 243 of file in6_var.h.

6.12.2.26 u_quad_t icmp6_ifstat::ifs6_out_msg

Definition at line 213 of file in6_var.h.

6.12.2.27 u_quad_t icmp6_ifstat::ifs6_out_neighboradvert

Definition at line 237 of file in6_var.h.

6.12.2.28 u_quad_t icmp6_ifstat::ifs6_out_neighborsolicit

Definition at line 235 of file in6_var.h.

6.12.2.29 u_quad_t icmp6_ifstat::ifs6_out_paramprob

Definition at line 223 of file in6_var.h.

6.12.2.30 u_quad_t icmp6_ifstat::ifs6_out_pkttoobig

Definition at line 225 of file in6_var.h.

6.12.2.31 u_quad_t icmp6_ifstat::ifs6_out_redirect

Definition at line 239 of file in6_var.h.

6.12.2.32 u_quad_t icmp6_ifstat::ifs6_out_routeradvert

Definition at line 233 of file in6_var.h.

6.12.2.33 u_quad_t icmp6_ifstat::ifs6_out_routersolicit

Definition at line 231 of file in6_var.h.

6.12.2.34 u_quad_t icmp6_ifstat::ifs6_out_timeexceed

Definition at line 221 of file in6_var.h.

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

- /usr/src/sys/netinet6/in6_var.h

6.13 if_set Struct Reference

```
#include <ip6_mroute.h>
```

Data Fields

- [if_mask ifs_bits](#) [howmany(IF_SETSIZE, NIFBITS)]

6.13.1 Detailed Description

Definition at line 89 of file ip6_mroute.h.

6.13.2 Field Documentation

6.13.2.1 [if_mask if_set::ifs_bits](#)[howmany(IF_SETSIZE, NIFBITS)]

Definition at line 90 of file ip6_mroute.h.

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

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.14 in6_addr Struct Reference

```
#include <in6.h>
```

Data Fields

- union {
 - uint8_t [__u6_addr8](#) [16]
 - uint16_t [__u6_addr16](#) [8]
 - uint32_t [__u6_addr32](#) [4]

6.14.1 Detailed Description

Definition at line 95 of file in6.h.

6.14.2 Field Documentation

6.14.2.1 union { ... } [in6_addr::__u6_addr](#)

6.14.2.2 uint16_t [in6_addr::__u6_addr16](#)[8]

Definition at line 98 of file in6.h.

6.14.2.3 uint32_t [in6_addr::__u6_addr32](#)[4]

Definition at line 99 of file in6.h.

6.14.2.4 uint8_t [in6_addr::__u6_addr8](#)[16]

Definition at line 97 of file in6.h.

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

- /usr/src/sys/netinet6/[in6.h](#)

6.15 in6_addrlifetime Struct Reference

```
#include <in6_var.h>
```

Data Fields

- [time_t ia6t_expire](#)
- [time_t ia6t_preferred](#)
- [u_int32_t ia6t_vltime](#)
- [u_int32_t ia6t_pltime](#)

6.15.1 Detailed Description

Definition at line 82 of file `in6_var.h`.

6.15.2 Field Documentation

6.15.2.1 `time_t in6_addrlifetime::ia6t_expire`

Definition at line 83 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_init_address_ltimes()`, `in6_update_ifa()`, and `ni6_store_addrs()`.

6.15.2.2 `u_int32_t in6_addrlifetime::ia6t_pltime`

Definition at line 86 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_init_address_ltimes()`, `in6_update_ifa()`, and `prelist_update()`.

6.15.2.3 `time_t in6_addrlifetime::ia6t_preferred`

Definition at line 84 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_init_address_ltimes()`, and `in6_update_ifa()`.

6.15.2.4 `u_int32_t in6_addrlifetime::ia6t_vltime`

Definition at line 85 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_init_address_ltimes()`, `in6_update_ifa()`, and `prelist_update()`.

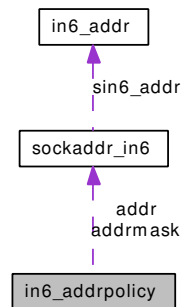
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6_var.h`

6.16 in6_addrpolicy Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_addrpolicy:



Data Fields

- [sockaddr_in6 addr](#)
- [sockaddr_in6 addrmask](#)
- [int preced](#)
- [int label](#)
- [u_quad_t use](#)

6.16.1 Detailed Description

Definition at line 121 of file in6_var.h.

6.16.2 Field Documentation

6.16.2.1 struct [sockaddr_in6](#) in6_addrpolicy::addr

Definition at line 122 of file in6_var.h.

Referenced by [in6_src_ioctl\(\)](#).

6.16.2.2 struct [sockaddr_in6](#) in6_addrpolicy::addrmask

Definition at line 123 of file in6_var.h.

Referenced by [in6_src_ioctl\(\)](#).

6.16.2.3 int in6_addrpolicy::label

Definition at line 125 of file in6_var.h.

Referenced by [in6_src_ioctl\(\)](#).

6.16.2.4 int [in6_addrpolicy::preced](#)

Definition at line 124 of file [in6_var.h](#).

6.16.2.5 u_quad_t [in6_addrpolicy::use](#)

Definition at line 126 of file [in6_var.h](#).

Referenced by [in6_src_ioctl\(\)](#), and [lookup_addrsel_policy\(\)](#).

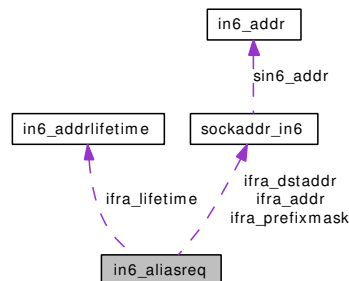
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.17 in6_aliasreq Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_aliasreq:



Data Fields

- char [ifra_name](#) [IFNAMSIZ]
- [sockaddr_in6 ifra_addr](#)
- [sockaddr_in6 ifra_dstaddr](#)
- [sockaddr_in6 ifra_prefixmask](#)
- int [ifra_flags](#)
- [in6_addrlifetime ifra_lifetime](#)

6.17.1 Detailed Description

Definition at line 264 of file in6_var.h.

6.17.2 Field Documentation

6.17.2.1 struct [sockaddr_in6](#) [in6_aliasreq::ifra_addr](#)

Definition at line 266 of file in6_var.h.

Referenced by [in6_control\(\)](#), and [in6_update_ifa\(\)](#).

6.17.2.2 struct [sockaddr_in6](#) [in6_aliasreq::ifra_dstaddr](#)

Definition at line 267 of file in6_var.h.

Referenced by [in6_update_ifa\(\)](#).

6.17.2.3 int [in6_aliasreq::ifra_flags](#)

Definition at line 269 of file in6_var.h.

Referenced by [in6_control\(\)](#), and [in6_update_ifa\(\)](#).

6.17.2.4 struct [in6_addrlifetime](#) [in6_aliasreq::ifra_lifetime](#)

Definition at line 270 of file [in6_var.h](#).

Referenced by [in6_control\(\)](#), and [in6_update_ifa\(\)](#).

6.17.2.5 char [in6_aliasreq::ifra_name](#)[IFNAMSIZ]

Definition at line 265 of file [in6_var.h](#).

6.17.2.6 struct [sockaddr_in6](#) [in6_aliasreq::ifra_prefixmask](#)

Definition at line 268 of file [in6_var.h](#).

Referenced by [in6_control\(\)](#), and [in6_update_ifa\(\)](#).

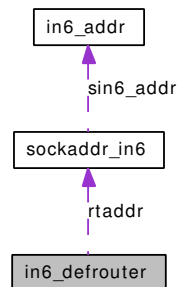
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.18 in6_defrouter Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_defrouter:



Data Fields

- `sockaddr_in6 rtaddr`
- `u_char flags`
- `u_short rtlifetime`
- `u_long expire`
- `u_short if_index`

6.18.1 Detailed Description

Definition at line 133 of file `nd6.h`.

6.18.2 Field Documentation

6.18.2.1 `u_long in6_defrouter::expire`

Definition at line 137 of file `nd6.h`.

6.18.2.2 `u_char in6_defrouter::flags`

Definition at line 135 of file `nd6.h`.

6.18.2.3 `u_short in6_defrouter::if_index`

Definition at line 138 of file `nd6.h`.

6.18.2.4 `struct sockaddr_in6 in6_defrouter::rtaddr`

Definition at line 134 of file `nd6.h`.

6.18.2.5 u_short [in6_defrouter::rtlifetime](#)

Definition at line 136 of file nd6.h.

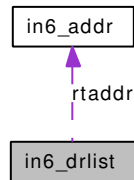
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.19 in6_drlist Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_drlist:



Data Fields

- char `ifname` [IFNAMSIZ]
- struct {
 - `in6_addr` `rtaddr`
 - `u_char` `flags`
 - `u_short` `rtlifetime`
 - `u_long` `expire`
 - `u_short` `if_index`
- } `defrouter` [DRLSTSIZ]

6.19.1 Detailed Description

Definition at line 122 of file nd6.h.

6.19.2 Field Documentation

6.19.2.1 struct { ... } `in6_drlist::defrouter`[DRLSTSIZ]

6.19.2.2 `u_long` `in6_drlist::expire`

Definition at line 128 of file nd6.h.

6.19.2.3 `u_char` `in6_drlist::flags`

Definition at line 126 of file nd6.h.

6.19.2.4 `u_short` `in6_drlist::if_index`

Definition at line 129 of file nd6.h.

6.19.2.5 `char` `in6_drlist::ifname`[IFNAMSIZ]

Definition at line 123 of file nd6.h.

6.19.2.6 struct [in6_addr](#) [in6_drlist::rtaddr](#)

Definition at line 125 of file [nd6.h](#).

6.19.2.7 u_short [in6_drlist::rtlifetime](#)

Definition at line 127 of file [nd6.h](#).

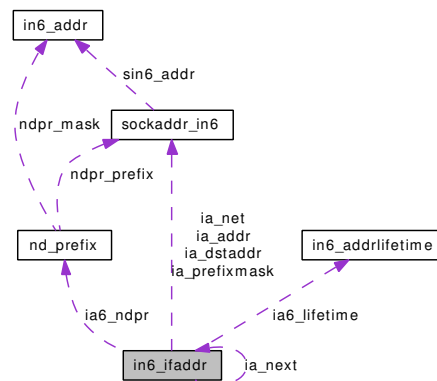
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.20 in6_ifaddr Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_ifaddr:



Data Fields

- ifaddr ia_ifa
- sockaddr_in6 ia_addr
- sockaddr_in6 ia_net
- sockaddr_in6 ia_dstaddr
- sockaddr_in6 ia_prefixmask
- u_int32_t ia_plen
- in6_ifaddr * ia_next
- int ia6_flags
- in6_addrlifetime ia6_lifetime
- time_t ia6_createtime
- time_t ia6_updatetime
- nd_prefix * ia6_ndpr

6.20.1 Detailed Description

Definition at line 98 of file in6_var.h.

6.20.2 Field Documentation

6.20.2.1 time_t in6_ifaddr::ia6_createtime

Definition at line 111 of file in6_var.h.

Referenced by prelist_update().

6.20.2.2 int [in6_ifaddr::ia6_flags](#)

Definition at line 108 of file `in6_var.h`.

Referenced by `icmp6_reflect()`, `in6_control()`, `in6_if_up()`, `in6_ifawithifp()`, `in6_is_addr_deprecated()`, `in6_lifaddr_ioctl()`, `in6_selectsrc()`, `in6_unlink_ifa()`, `in6_update_ifa()`, `ip6_input()`, `nd6_dad_duplicated()`, `nd6_dad_start()`, `nd6_dad_timer()`, `nd6_ioctl()`, `nd6_ns_input()`, `nd6_timer()`, `ni6_addrs()`, `ni6_input()`, `ni6_store_addrs()`, `pfxlist_onlink_check()`, `prelist_update()`, and `regen_tmpaddr()`.

6.20.2.3 struct [in6_addrlifetime in6_ifaddr::ia6_lifetime](#)

Definition at line 110 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_update_ifa()`, `nd6_timer()`, `ni6_store_addrs()`, and `prelist_update()`.

6.20.2.4 struct [nd_prefix* in6_ifaddr::ia6_ndpr](#)

Definition at line 117 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_unlink_ifa()`, `nd6_ioctl()`, `pfxlist_onlink_check()`, `prelist_update()`, and `regen_tmpaddr()`.

6.20.2.5 time_t [in6_ifaddr::ia6_updatetime](#)

Definition at line 114 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_update_ifa()`, and `prelist_update()`.

6.20.2.6 struct [sockaddr_in6 in6_ifaddr::ia_addr](#)

Definition at line 102 of file `in6_var.h`.

Referenced by `icmp6_redirect_output()`, `icmp6_reflect()`, `in6_control()`, `in6_ifadd()`, `in6_ifdetach()`, `in6_ifinit()`, `in6_ifremloop()`, `in6_is_addr_deprecated()`, `in6_lifaddr_ioctl()`, `in6_localaddr()`, `in6_purgeaddr()`, `in6_selectsrc()`, `in6_update_ifa()`, `mld6_sendpkt()`, `nd6_dad_duplicated()`, `nd6_dad_ns_input()`, `nd6_dad_ns_output()`, `nd6_dad_start()`, `nd6_dad_timer()`, `ni6_addrs()`, and `ni6_store_addrs()`.

6.20.2.7 struct [sockaddr_in6 in6_ifaddr::ia_dstaddr](#)

Definition at line 104 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_ifinit()`, `in6_lifaddr_ioctl()`, `in6_purgeaddr()`, and `in6_update_ifa()`.

6.20.2.8 struct [ifaddr in6_ifaddr::ia_ifa](#)

Definition at line 99 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_ifdetach()`, `in6_ifinit()`, `in6_purgeaddr()`, `in6_unlink_ifa()`, `in6_update_ifa()`, `ip6_input()`, `ip6_output()`, `nd6_ioctl()`, and `nd6_timer()`.

6.20.2.9 struct [sockaddr_in6 in6_ifaddr::ia_net](#)

Definition at line 103 of file `in6_var.h`.

6.20.2.10 struct `in6_ifaddr*` `in6_ifaddr::ia_next`

Definition at line 107 of file `in6_var.h`.

Referenced by `in6_ifdetach()`, `in6_ifremloop()`, `in6_is_addr_deprecated()`, `in6_localaddr()`, `in6_selectsrc()`, `in6_unlink_ifa()`, `in6_update_ifa()`, `nd6_ioctl()`, `nd6_timer()`, and `pfxlist_onlink_check()`.

6.20.2.11 u_int32_t `in6_ifaddr::ia_plen`

Definition at line 106 of file `in6_var.h`.

6.20.2.12 struct `sockaddr_in6` `in6_ifaddr::ia_prefixmask`

Definition at line 105 of file `in6_var.h`.

Referenced by `in6_control()`, `in6_ifadd()`, `in6_ifdetach()`, `in6_ifinit()`, `in6_lifaddr_ioctl()`, `in6_localaddr()`, and `in6_update_ifa()`.

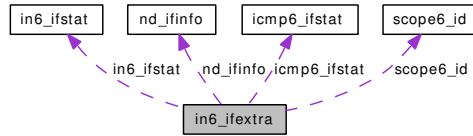
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6_var.h`

6.21 in6_ifextra Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_ifextra:



Data Fields

- [in6_ifstat](#) * [in6_ifstat](#)
- [icmp6_ifstat](#) * [icmp6_ifstat](#)
- [nd_ifinfo](#) * [nd_ifinfo](#)
- [scope6_id](#) * [scope6_id](#)

6.21.1 Detailed Description

Definition at line 91 of file in6_var.h.

6.21.2 Field Documentation

6.21.2.1 struct [icmp6_ifstat](#)* [in6_ifextra::icmp6_ifstat](#)

Definition at line 93 of file in6_var.h.

Referenced by [in6_domifdetach\(\)](#).

6.21.2.2 struct [in6_ifstat](#)* [in6_ifextra::in6_ifstat](#)

Definition at line 92 of file in6_var.h.

Referenced by [in6_domifdetach\(\)](#).

6.21.2.3 struct [nd_ifinfo](#)* [in6_ifextra::nd_ifinfo](#)

Definition at line 94 of file in6_var.h.

Referenced by [in6_domifdetach\(\)](#).

6.21.2.4 struct [scope6_id](#)* [in6_ifextra::scope6_id](#)

Definition at line 95 of file in6_var.h.

Referenced by [in6_domifdetach\(\)](#).

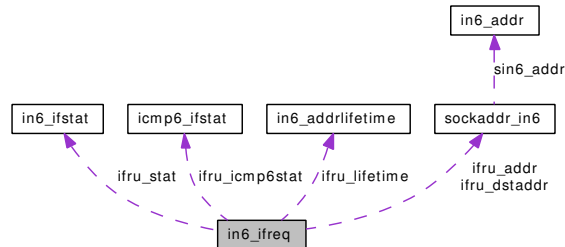
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.22 in6_ifreq Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_ifreq:



Data Fields

- char `ifr_name` [IFNAMSIZ]
- union {
 - `sockaddr_in6 ifru_addr`
 - `sockaddr_in6 ifru_dstaddr`
 - int `ifru_flags`
 - int `ifru_flags6`
 - int `ifru_metric`
 - `caddr_t ifru_data`
 - `in6_addrlifetime ifru_lifetime`
 - `in6_ifstat ifru_stat`
 - `icmp6_ifstat ifru_icmp6stat`
 - `u_int32_t ifru_scope_id` [16]
- } `ifr_ifru`

6.22.1 Detailed Description

Definition at line 248 of file `in6_var.h`.

6.22.2 Field Documentation

6.22.2.1 union { ... } `in6_ifreq::ifr_ifru`

Referenced by `in6_control()`.

6.22.2.2 char `in6_ifreq::ifr_name`[IFNAMSIZ]

Definition at line 249 of file `in6_var.h`.

6.22.2.3 struct `sockaddr_in6` `in6_ifreq::ifru_addr`

Definition at line 251 of file `in6_var.h`.

6.22.2.4 `caddr_t in6_ifreq::ifru_data`

Definition at line 256 of file `in6_var.h`.

6.22.2.5 `struct sockaddr_in6 in6_ifreq::ifru_dstaddr`

Definition at line 252 of file `in6_var.h`.

6.22.2.6 `int in6_ifreq::ifru_flags`

Definition at line 253 of file `in6_var.h`.

6.22.2.7 `int in6_ifreq::ifru_flags6`

Definition at line 254 of file `in6_var.h`.

Referenced by `in6_control()`.

6.22.2.8 `struct icmp6_ifstat in6_ifreq::ifru_icmp6stat`

Definition at line 259 of file `in6_var.h`.

Referenced by `in6_control()`.

6.22.2.9 `struct in6_addrlifetime in6_ifreq::ifru_lifetime`

Definition at line 257 of file `in6_var.h`.

Referenced by `in6_control()`.

6.22.2.10 `int in6_ifreq::ifru_metric`

Definition at line 255 of file `in6_var.h`.

6.22.2.11 `u_int32_t in6_ifreq::ifru_scope_id[16]`

Definition at line 260 of file `in6_var.h`.

Referenced by `in6_control()`.

6.22.2.12 `struct in6_ifstat in6_ifreq::ifru_stat`

Definition at line 258 of file `in6_var.h`.

Referenced by `in6_control()`.

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

- `/usr/src/sys/netinet6/in6_var.h`

6.23 in6_ifstat Struct Reference

```
#include <in6_var.h>
```

Data Fields

- [u_quad_t ifs6_in_receive](#)
- [u_quad_t ifs6_in_hdrerr](#)
- [u_quad_t ifs6_in_toobig](#)
- [u_quad_t ifs6_in_noroute](#)
- [u_quad_t ifs6_in_addrerr](#)
- [u_quad_t ifs6_in_protunknown](#)
- [u_quad_t ifs6_in_truncated](#)
- [u_quad_t ifs6_in_discard](#)
- [u_quad_t ifs6_in_deliver](#)
- [u_quad_t ifs6_out_forward](#)
- [u_quad_t ifs6_out_request](#)
- [u_quad_t ifs6_out_discard](#)
- [u_quad_t ifs6_out_fragok](#)
- [u_quad_t ifs6_out_fragfail](#)
- [u_quad_t ifs6_out_fragcreat](#)
- [u_quad_t ifs6_reass_reqd](#)
- [u_quad_t ifs6_reass_ok](#)
- [u_quad_t ifs6_reass_fail](#)
- [u_quad_t ifs6_in_mcast](#)
- [u_quad_t ifs6_out_mcast](#)

6.23.1 Detailed Description

Definition at line 132 of file in6_var.h.

6.23.2 Field Documentation

6.23.2.1 [u_quad_t in6_ifstat::ifs6_in_addrerr](#)

Definition at line 137 of file in6_var.h.

6.23.2.2 [u_quad_t in6_ifstat::ifs6_in_deliver](#)

Definition at line 143 of file in6_var.h.

6.23.2.3 [u_quad_t in6_ifstat::ifs6_in_discard](#)

Definition at line 141 of file in6_var.h.

6.23.2.4 [u_quad_t in6_ifstat::ifs6_in_hdrerr](#)

Definition at line 134 of file in6_var.h.

6.23.2.5 u_quad_t in6_ifstat::ifs6_in_mcast

Definition at line 162 of file in6_var.h.

6.23.2.6 u_quad_t in6_ifstat::ifs6_in_noroute

Definition at line 136 of file in6_var.h.

6.23.2.7 u_quad_t in6_ifstat::ifs6_in_protunknown

Definition at line 138 of file in6_var.h.

6.23.2.8 u_quad_t in6_ifstat::ifs6_in_receive

Definition at line 133 of file in6_var.h.

6.23.2.9 u_quad_t in6_ifstat::ifs6_in_toobig

Definition at line 135 of file in6_var.h.

6.23.2.10 u_quad_t in6_ifstat::ifs6_in_truncated

Definition at line 140 of file in6_var.h.

6.23.2.11 u_quad_t in6_ifstat::ifs6_out_discard

Definition at line 149 of file in6_var.h.

6.23.2.12 u_quad_t in6_ifstat::ifs6_out_forward

Definition at line 145 of file in6_var.h.

6.23.2.13 u_quad_t in6_ifstat::ifs6_out_fragreat

Definition at line 152 of file in6_var.h.

6.23.2.14 u_quad_t in6_ifstat::ifs6_out_fragfail

Definition at line 151 of file in6_var.h.

6.23.2.15 u_quad_t in6_ifstat::ifs6_out_fragok

Definition at line 150 of file in6_var.h.

6.23.2.16 `u_quad_t in6_ifstat::ifs6_out_mcast`

Definition at line 163 of file `in6_var.h`.

6.23.2.17 `u_quad_t in6_ifstat::ifs6_out_request`

Definition at line 147 of file `in6_var.h`.

6.23.2.18 `u_quad_t in6_ifstat::ifs6_reass_fail`

Definition at line 159 of file `in6_var.h`.

6.23.2.19 `u_quad_t in6_ifstat::ifs6_reass_ok`

Definition at line 156 of file `in6_var.h`.

6.23.2.20 `u_quad_t in6_ifstat::ifs6_reass_reqd`

Definition at line 154 of file `in6_var.h`.

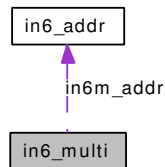
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6_var.h`

6.24 in6_multi Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_multi:



Public Member Functions

- [LIST_ENTRY](#) ([in6_multi](#)) [in6m_entry](#)

Data Fields

- [in6_addr](#) [in6m_addr](#)
- [ifnet *](#) [in6m_ifp](#)
- [ifmultiaddr *](#) [in6m_ifma](#)
- [u_int](#) [in6m_refcount](#)
- [u_int](#) [in6m_state](#)
- [u_int](#) [in6m_timer](#)
- [timeval](#) [in6m_timer_expire](#)
- [callout *](#) [in6m_timer_ch](#)

6.24.1 Detailed Description

Definition at line 514 of file [in6_var.h](#).

6.24.2 Member Function Documentation

6.24.2.1 [in6_multi::LIST_ENTRY](#) ([in6_multi](#))

6.24.3 Field Documentation

6.24.3.1 [struct in6_addr](#) [in6_multi::in6m_addr](#)

Definition at line 516 of file [in6_var.h](#).

Referenced by [ip6_setmoptions\(\)](#), and [mld6_input\(\)](#).

6.24.3.2 [struct ifmultiaddr*](#) [in6_multi::in6m_ifma](#)

Definition at line 518 of file [in6_var.h](#).

6.24.3.3 struct ifnet* [in6_multi::in6m_ifp](#)

Definition at line 517 of file [in6_var.h](#).

Referenced by [in6_ifdetach\(\)](#), [in6_pcbpurgeif0\(\)](#), and [ip6_setmoptions\(\)](#).

6.24.3.4 u_int [in6_multi::in6m_refcount](#)

Definition at line 519 of file [in6_var.h](#).

6.24.3.5 u_int [in6_multi::in6m_state](#)

Definition at line 520 of file [in6_var.h](#).

Referenced by [in6_update_ifa\(\)](#), and [mld6_input\(\)](#).

6.24.3.6 u_int [in6_multi::in6m_timer](#)

Definition at line 521 of file [in6_var.h](#).

Referenced by [in6_update_ifa\(\)](#), and [mld6_input\(\)](#).

6.24.3.7 struct callout* [in6_multi::in6m_timer_ch](#)

Definition at line 523 of file [in6_var.h](#).

6.24.3.8 struct timeval [in6_multi::in6m_timer_expire](#)

Definition at line 522 of file [in6_var.h](#).

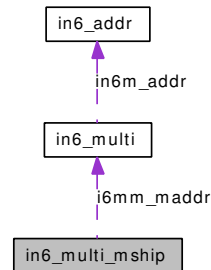
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.25 in6_multi_mship Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_multi_mship:



Public Member Functions

- `LIST_ENTRY` (`in6_multi_mship`) `i6mm_chain`

Data Fields

- `in6_multi * i6mm_maddr`

6.25.1 Detailed Description

Definition at line 509 of file `in6_var.h`.

6.25.2 Member Function Documentation

6.25.2.1 `in6_multi_mship::LIST_ENTRY` (`in6_multi_mship`)

6.25.3 Field Documentation

6.25.3.1 `struct in6_multi* in6_multi_mship::i6mm_maddr`

Definition at line 510 of file `in6_var.h`.

Referenced by `in6_pcbpurgeif0()`, `in6_update_ifa()`, and `ip6_setmoptions()`.

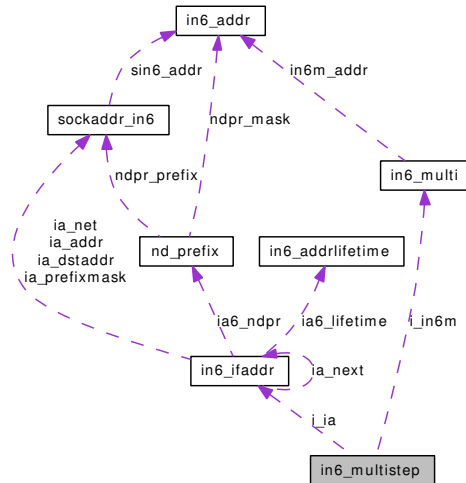
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6_var.h`

6.26 in6_multistep Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_multistep:



Data Fields

- [in6_ifaddr](#) * [i_ia](#)
- [in6_multi](#) * [i_in6m](#)

6.26.1 Detailed Description

Definition at line 538 of file in6_var.h.

6.26.2 Field Documentation

6.26.2.1 struct [in6_ifaddr](#)* [in6_multistep::i_ia](#)

Definition at line 539 of file in6_var.h.

6.26.2.2 struct [in6_multi](#)* [in6_multistep::i_in6m](#)

Definition at line 540 of file in6_var.h.

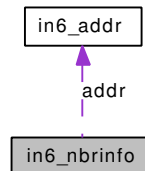
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.27 in6_nbrinfo Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_nbrinfo:



Data Fields

- char `ifname` [IFNAMSIZ]
- `in6_addr` `addr`
- long `asked`
- int `isrouter`
- int `state`
- int `expire`

6.27.1 Detailed Description

Definition at line 111 of file `nd6.h`.

6.27.2 Field Documentation

6.27.2.1 struct `in6_addr` `in6_nbrinfo::addr`

Definition at line 113 of file `nd6.h`.

Referenced by `nd6_ioctl()`.

6.27.2.2 long `in6_nbrinfo::asked`

Definition at line 114 of file `nd6.h`.

Referenced by `nd6_ioctl()`.

6.27.2.3 int `in6_nbrinfo::expire`

Definition at line 117 of file `nd6.h`.

Referenced by `nd6_ioctl()`.

6.27.2.4 char `in6_nbrinfo::ifname`[IFNAMSIZ]

Definition at line 112 of file `nd6.h`.

6.27.2.5 int [in6_nbrinfo::isrouter](#)

Definition at line 115 of file nd6.h.

Referenced by [nd6_ioctl\(\)](#).

6.27.2.6 int [in6_nbrinfo::state](#)

Definition at line 116 of file nd6.h.

Referenced by [nd6_ioctl\(\)](#).

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

- [/usr/src/sys/netinet6/nd6.h](#)

6.28 in6_ndifreq Struct Reference

```
#include <nd6.h>
```

Data Fields

- char [ifname](#) [IFNAMSIZ]
- u_long [ifindex](#)

6.28.1 Detailed Description

Definition at line 212 of file nd6.h.

6.28.2 Field Documentation

6.28.2.1 u_long [in6_ndifreq::ifindex](#)

Definition at line 214 of file nd6.h.

Referenced by [nd6_ioctl\(\)](#).

6.28.2.2 char [in6_ndifreq::ifname](#)[IFNAMSIZ]

Definition at line 213 of file nd6.h.

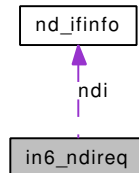
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.29 in6_ndireq Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_ndireq:



Data Fields

- char [ifname](#) [IFNAMSIZ]
- [nd_ifinfo](#) ndi

6.29.1 Detailed Description

Definition at line 207 of file nd6.h.

6.29.2 Field Documentation

6.29.2.1 char [in6_ndireq::ifname](#)[IFNAMSIZ]

Definition at line 208 of file nd6.h.

6.29.2.2 struct [nd_ifinfo](#) [in6_ndireq::ndi](#)

Definition at line 209 of file nd6.h.

Referenced by [nd6_ioctl\(\)](#).

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

- [/usr/src/sys/netinet6/nd6.h](#)

6.30 in6_ondireq Struct Reference

```
#include <nd6.h>
```

Data Fields

- char [ifname](#) [IFNAMSIZ]
- struct {
 - u_int32_t [linkmtu](#)
 - u_int32_t [maxmtu](#)
 - u_int32_t [basereachable](#)
 - u_int32_t [reachable](#)
 - u_int32_t [retrans](#)
 - u_int32_t [flags](#)
 - int [recalctm](#)
 - u_int8_t [chlim](#)
 - u_int8_t [receivedra](#)

6.30.1 Detailed Description

Definition at line 191 of file nd6.h.

6.30.2 Field Documentation

6.30.2.1 u_int32_t [in6_ondireq::basereachable](#)

Definition at line 196 of file nd6.h.

6.30.2.2 u_int8_t [in6_ondireq::chlim](#)

Definition at line 201 of file nd6.h.

6.30.2.3 u_int32_t [in6_ondireq::flags](#)

Definition at line 199 of file nd6.h.

6.30.2.4 char [in6_ondireq::ifname](#)[IFNAMSIZ]

Definition at line 192 of file nd6.h.

6.30.2.5 u_int32_t [in6_ondireq::linkmtu](#)

Definition at line 194 of file nd6.h.

6.30.2.6 `u_int32_t in6_ondireq::maxmtu`

Definition at line 195 of file nd6.h.

6.30.2.7 `struct { ... } in6_ondireq::ndi`**6.30.2.8** `u_int32_t in6_ondireq::reachable`

Definition at line 197 of file nd6.h.

6.30.2.9 `int in6_ondireq::recalctm`

Definition at line 200 of file nd6.h.

6.30.2.10 `u_int8_t in6_ondireq::receivedra`

Definition at line 202 of file nd6.h.

6.30.2.11 `u_int32_t in6_ondireq::retrans`

Definition at line 198 of file nd6.h.

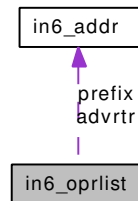
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.31 in6_oprlist Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_oprlist:



Data Fields

- char [ifname](#) [IFNAMSIZ]
- struct {
 - [in6_addr](#) [prefix](#)
 - [prf_ra](#) [raflags](#)
 - [u_char](#) [prefixlen](#)
 - [u_char](#) [origin](#)
 - [u_long](#) [vlttime](#)
 - [u_long](#) [ptime](#)
 - [u_long](#) [expire](#)
 - [u_short](#) [if_index](#)
 - [u_short](#) [advrtrs](#)
 - [in6_addr](#) [advrtr](#) [DRLSTSIZ]
 - } [prefix](#) [PRLSTSIZ]

6.31.1 Detailed Description

Definition at line 142 of file nd6.h.

6.31.2 Field Documentation

6.31.2.1 struct [in6_addr](#) [in6_oprlist::advrtr](#)[DRLSTSIZ]

Definition at line 154 of file nd6.h.

6.31.2.2 [u_short](#) [in6_oprlist::advrtrs](#)

Definition at line 153 of file nd6.h.

6.31.2.3 [u_long](#) [in6_oprlist::expire](#)

Definition at line 151 of file nd6.h.

6.31.2.4 `u_short in6_oprlist::if_index`

Definition at line 152 of file nd6.h.

6.31.2.5 `char in6_oprlist::ifname[IFNAMSIZ]`

Definition at line 143 of file nd6.h.

6.31.2.6 `u_char in6_oprlist::origin`

Definition at line 148 of file nd6.h.

6.31.2.7 `u_long in6_oprlist::ptime`

Definition at line 150 of file nd6.h.

6.31.2.8 `struct { ... } in6_oprlist::prefix[PRLSTSIZ]`**6.31.2.9** `struct in6_addr in6_oprlist::prefix`

Definition at line 145 of file nd6.h.

6.31.2.10 `u_char in6_oprlist::prefixlen`

Definition at line 147 of file nd6.h.

6.31.2.11 `struct prf_ra in6_oprlist::raflags`

Definition at line 146 of file nd6.h.

6.31.2.12 `u_long in6_oprlist::vtime`

Definition at line 149 of file nd6.h.

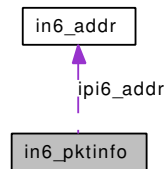
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/nd6.h`

6.32 in6_pktinfo Struct Reference

```
#include <in6.h>
```

Collaboration diagram for in6_pktinfo:



Data Fields

- [in6_addr ipi6_addr](#)
- unsigned int [ipi6_ifindex](#)

6.32.1 Detailed Description

Definition at line 493 of file in6.h.

6.32.2 Field Documentation

6.32.2.1 struct [in6_addr](#) [in6_pktinfo::ipi6_addr](#)

Definition at line 494 of file in6.h.

Referenced by [in6_selectsrc\(\)](#), [ip6_savecontrol\(\)](#), and [ip6_setpktopt\(\)](#).

6.32.2.2 unsigned int [in6_pktinfo::ipi6_ifindex](#)

Definition at line 495 of file in6.h.

Referenced by [ip6_savecontrol\(\)](#), [ip6_setpktopt\(\)](#), and [selectroute\(\)](#).

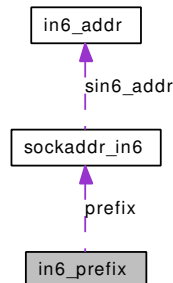
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6.h](#)

6.33 in6_prefix Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_prefix:



Data Fields

- [sockaddr_in6 prefix](#)
- [prf_ra raflags](#)
- [u_char prefixlen](#)
- [u_char origin](#)
- [u_int32_t vlttime](#)
- [u_int32_t pltime](#)
- [time_t expire](#)
- [u_int32_t flags](#)
- [int refcnt](#)
- [u_short if_index](#)
- [u_short advrtrs](#)

6.33.1 Detailed Description

Definition at line 175 of file nd6.h.

6.33.2 Field Documentation

6.33.2.1 u_short in6_prefix::advrtrs

Definition at line 186 of file nd6.h.

6.33.2.2 time_t in6_prefix::expire

Definition at line 182 of file nd6.h.

6.33.2.3 u_int32_t in6_prefix::flags

Definition at line 183 of file nd6.h.

6.33.2.4 u_short in6_prefix::if_index

Definition at line 185 of file nd6.h.

6.33.2.5 u_char in6_prefix::origin

Definition at line 179 of file nd6.h.

6.33.2.6 u_int32_t in6_prefix::pltime

Definition at line 181 of file nd6.h.

6.33.2.7 struct sockaddr_in6 in6_prefix::prefix

Definition at line 176 of file nd6.h.

6.33.2.8 u_char in6_prefix::prefixlen

Definition at line 178 of file nd6.h.

6.33.2.9 struct prf_ra in6_prefix::raflags

Definition at line 177 of file nd6.h.

6.33.2.10 int in6_prefix::refcnt

Definition at line 184 of file nd6.h.

6.33.2.11 u_int32_t in6_prefix::vlttime

Definition at line 180 of file nd6.h.

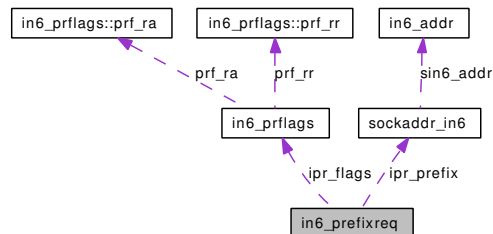
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.34 in6_prefixreq Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_prefixreq:



Data Fields

- char `ipr_name` [IFNAMSIZ]
- u_char `ipr_origin`
- u_char `ipr_plen`
- u_int32_t `ipr_vltime`
- u_int32_t `ipr_pltime`
- `in6_prflags` `ipr_flags`
- `sockaddr_in6` `ipr_prefix`

6.34.1 Detailed Description

Definition at line 299 of file `in6_var.h`.

6.34.2 Field Documentation

6.34.2.1 struct `in6_prflags` `in6_prefixreq::ipr_flags`

Definition at line 305 of file `in6_var.h`.

6.34.2.2 char `in6_prefixreq::ipr_name`[IFNAMSIZ]

Definition at line 300 of file `in6_var.h`.

6.34.2.3 u_char `in6_prefixreq::ipr_origin`

Definition at line 301 of file `in6_var.h`.

6.34.2.4 u_char `in6_prefixreq::ipr_plen`

Definition at line 302 of file `in6_var.h`.

6.34.2.5 `u_int32_t in6_prefixreq::ipr_pltime`

Definition at line 304 of file `in6_var.h`.

6.34.2.6 `struct sockaddr_in6 in6_prefixreq::ipr_prefix`

Definition at line 306 of file `in6_var.h`.

6.34.2.7 `u_int32_t in6_prefixreq::ipr_vltime`

Definition at line 303 of file `in6_var.h`.

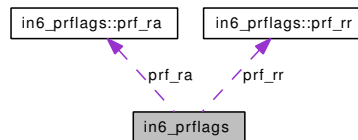
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_var.h](#)

6.35 in6_prflags Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_prflags:



Data Fields

- [in6_prflags::prf_ra](#) [prf_ra](#)
- [u_char](#) [prf_reserved1](#)
- [u_short](#) [prf_reserved2](#)
- [in6_prflags::prf_rr](#) [prf_rr](#)
- [u_char](#) [prf_reserved3](#)
- [u_short](#) [prf_reserved4](#)

Data Structures

- [struct](#) [prf_ra](#)
- [struct](#) [prf_rr](#)

6.35.1 Detailed Description

Definition at line 281 of file in6_var.h.

6.35.2 Field Documentation

6.35.2.1 [struct](#) [in6_prflags::prf_ra](#) [in6_prflags::prf_ra](#)

6.35.2.2 [u_char](#) [in6_prflags::prf_reserved1](#)

Definition at line 287 of file in6_var.h.

6.35.2.3 [u_short](#) [in6_prflags::prf_reserved2](#)

Definition at line 288 of file in6_var.h.

6.35.2.4 [u_char](#) [in6_prflags::prf_reserved3](#)

Definition at line 295 of file in6_var.h.

6.35.2.5 `u_short in6_prflags::prf_reserved4`

Definition at line 296 of file `in6_var.h`.

6.35.2.6 `struct in6_prflags::prf_rr in6_prflags::prf_rr`

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

- `/usr/src/sys/netinet6/in6_var.h`

6.36 in6_prflags::prf_ra Struct Reference

```
#include <in6_var.h>
```

Data Fields

- u_char [onlink](#): 1
- u_char [autonomous](#): 1
- u_char [reserved](#): 6

6.36.1 Detailed Description

Definition at line 282 of file in6_var.h.

6.36.2 Field Documentation

6.36.2.1 u_char [in6_prflags::prf_ra::autonomous](#)

Definition at line 284 of file in6_var.h.

6.36.2.2 u_char [in6_prflags::prf_ra::onlink](#)

Definition at line 283 of file in6_var.h.

6.36.2.3 u_char [in6_prflags::prf_ra::reserved](#)

Definition at line 285 of file in6_var.h.

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

- /usr/src/sys/netinet6/in6_var.h

6.37 in6_prflags::prf_rr Struct Reference

```
#include <in6_var.h>
```

Data Fields

- u_char [decrvalid](#): 1
- u_char [decrprefd](#): 1
- u_char [reserved](#): 6

6.37.1 Detailed Description

Definition at line 290 of file in6_var.h.

6.37.2 Field Documentation

6.37.2.1 u_char [in6_prflags::prf_rr::decrprefd](#)

Definition at line 292 of file in6_var.h.

6.37.2.2 u_char [in6_prflags::prf_rr::decrvalid](#)

Definition at line 291 of file in6_var.h.

6.37.2.3 u_char [in6_prflags::prf_rr::reserved](#)

Definition at line 293 of file in6_var.h.

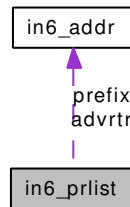
The documentation for this struct was generated from the following file:

- /usr/src/sys/netinet6/in6_var.h

6.38 in6_prlist Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for in6_prlist:



Data Fields

- char [ifname](#) [IFNAMSIZ]
- struct {
 - [in6_addr](#) [prefix](#)
 - prf_ra [raflags](#)
 - u_char [prefixlen](#)
 - u_char [origin](#)
 - u_int32_t [vtime](#)
 - u_int32_t [ptime](#)
 - time_t [expire](#)
 - u_short [if_index](#)
 - u_short [advrtrs](#)
 - [in6_addr](#) [advrtr](#) [DRLSTSIZ]
 - } [prefix](#) [PRLSTSIZ]

6.38.1 Detailed Description

Definition at line 159 of file nd6.h.

6.38.2 Field Documentation

6.38.2.1 struct [in6_addr](#) [in6_prlist::advrtr](#)[DRLSTSIZ]

Definition at line 171 of file nd6.h.

6.38.2.2 u_short [in6_prlist::advrtrs](#)

Definition at line 170 of file nd6.h.

6.38.2.3 time_t [in6_prlist::expire](#)

Definition at line 168 of file nd6.h.

6.38.2.4 u_short in6_prlist::if_index

Definition at line 169 of file nd6.h.

6.38.2.5 char in6_prlist::ifname[IFNAMSIZ]

Definition at line 160 of file nd6.h.

6.38.2.6 u_char in6_prlist::origin

Definition at line 165 of file nd6.h.

6.38.2.7 u_int32_t in6_prlist::ptime

Definition at line 167 of file nd6.h.

6.38.2.8 struct { ... } in6_prlist::prefix[PRLSTSIZ]**6.38.2.9 struct in6_addr in6_prlist::prefix**

Definition at line 162 of file nd6.h.

6.38.2.10 u_char in6_prlist::prefixlen

Definition at line 164 of file nd6.h.

6.38.2.11 struct prf_ra in6_prlist::raflags

Definition at line 163 of file nd6.h.

6.38.2.12 u_int32_t in6_prlist::vtime

Definition at line 166 of file nd6.h.

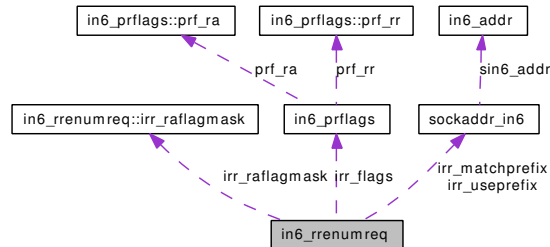
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.39 in6_rrenumreq Struct Reference

```
#include <in6_var.h>
```

Collaboration diagram for in6_rrenumreq:



Data Fields

- char [irr_name](#) [IFNAMSIZ]
- u_char [irr_origin](#)
- u_char [irr_m_len](#)
- u_char [irr_m_minlen](#)
- u_char [irr_m_maxlen](#)
- u_char [irr_u_uselen](#)
- u_char [irr_u_keeplen](#)
- [in6_rrenumreq::irr_raflagmask](#) irr_raflagmask
- u_int32_t [irr_vltime](#)
- u_int32_t [irr_pltime](#)
- [in6_prflags](#) irr_flags
- [sockaddr_in6](#) irr_matchprefix
- [sockaddr_in6](#) irr_useprefix

Data Structures

- struct [irr_raflagmask](#)

6.39.1 Detailed Description

Definition at line 322 of file in6_var.h.

6.39.2 Field Documentation

6.39.2.1 struct [in6_prflags](#) [in6_rrenumreq::irr_flags](#)

Definition at line 337 of file in6_var.h.

6.39.2.2 u_char [in6_rrenumreq::irr_m_len](#)

Definition at line 325 of file in6_var.h.

6.39.2.3 u_char in6_rrenumreq::irr_m_maxlen

Definition at line 327 of file in6_var.h.

6.39.2.4 u_char in6_rrenumreq::irr_m_minlen

Definition at line 326 of file in6_var.h.

6.39.2.5 struct sockaddr_in6 in6_rrenumreq::irr_matchprefix

Definition at line 338 of file in6_var.h.

6.39.2.6 char in6_rrenumreq::irr_name[IFNAMSIZ]

Definition at line 323 of file in6_var.h.

6.39.2.7 u_char in6_rrenumreq::irr_origin

Definition at line 324 of file in6_var.h.

6.39.2.8 u_int32_t in6_rrenumreq::irr_pltime

Definition at line 336 of file in6_var.h.

6.39.2.9 struct in6_rrenumreq::irr_raflagmask in6_rrenumreq::irr_raflagmask**6.39.2.10 u_char in6_rrenumreq::irr_u_keeplen**

Definition at line 329 of file in6_var.h.

6.39.2.11 u_char in6_rrenumreq::irr_u_uselen

Definition at line 328 of file in6_var.h.

6.39.2.12 struct sockaddr_in6 in6_rrenumreq::irr_useprefix

Definition at line 339 of file in6_var.h.

6.39.2.13 u_int32_t in6_rrenumreq::irr_vltime

Definition at line 335 of file in6_var.h.

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

- /usr/src/sys/netinet6/in6_var.h

6.40 in6_rrenumreq::irr_raflagmask Struct Reference

```
#include <in6_var.h>
```

Data Fields

- u_char [onlink](#): 1
- u_char [autonomous](#): 1
- u_char [reserved](#): 6

6.40.1 Detailed Description

Definition at line 330 of file in6_var.h.

6.40.2 Field Documentation

6.40.2.1 u_char [in6_rrenumreq::irr_raflagmask::autonomous](#)

Definition at line 332 of file in6_var.h.

6.40.2.2 u_char [in6_rrenumreq::irr_raflagmask::onlink](#)

Definition at line 331 of file in6_var.h.

6.40.2.3 u_char [in6_rrenumreq::irr_raflagmask::reserved](#)

Definition at line 333 of file in6_var.h.

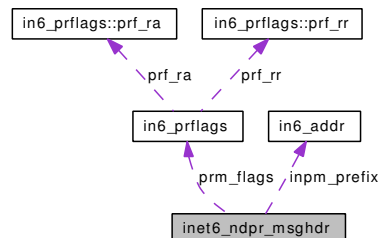
The documentation for this struct was generated from the following file:

- /usr/src/sys/netinet6/in6_var.h

6.41 inet6_ndpr_msghdr Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for inet6_ndpr_msghdr:



Data Fields

- u_short [inpm_msglen](#)
- u_char [inpm_version](#)
- u_char [inpm_type](#)
- [in6_addr](#) [inpm_prefix](#)
- u_long [prm_vltim](#)
- u_long [prm_pltime](#)
- u_long [prm_expire](#)
- u_long [prm_preferred](#)
- [in6_prflags](#) [prm_flags](#)
- u_short [prm_index](#)
- u_char [prm_plen](#)

6.41.1 Detailed Description

Definition at line 300 of file nd6.h.

6.41.2 Field Documentation

6.41.2.1 u_short [inet6_ndpr_msghdr::inpm_msglen](#)

Definition at line 301 of file nd6.h.

6.41.2.2 struct [in6_addr](#) [inet6_ndpr_msghdr::inpm_prefix](#)

Definition at line 304 of file nd6.h.

6.41.2.3 u_char [inet6_ndpr_msghdr::inpm_type](#)

Definition at line 303 of file nd6.h.

6.41.2.4 `u_char` [inet6_ndpr_msghdr::inpm_version](#)

Definition at line 302 of file nd6.h.

6.41.2.5 `u_long` [inet6_ndpr_msghdr::prm_expire](#)

Definition at line 307 of file nd6.h.

6.41.2.6 `struct in6_prflags` [inet6_ndpr_msghdr::prm_flags](#)

Definition at line 309 of file nd6.h.

6.41.2.7 `u_short` [inet6_ndpr_msghdr::prm_index](#)

Definition at line 310 of file nd6.h.

6.41.2.8 `u_char` [inet6_ndpr_msghdr::prm_plen](#)

Definition at line 311 of file nd6.h.

6.41.2.9 `u_long` [inet6_ndpr_msghdr::prm_pltime](#)

Definition at line 306 of file nd6.h.

6.41.2.10 `u_long` [inet6_ndpr_msghdr::prm_preferred](#)

Definition at line 308 of file nd6.h.

6.41.2.11 `u_long` [inet6_ndpr_msghdr::prm_vltim](#)

Definition at line 305 of file nd6.h.

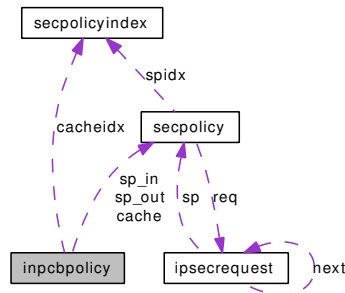
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.42 inpcbpolicy Struct Reference

```
#include <ipsec.h>
```

Collaboration diagram for inpcbpolicy:



Data Fields

- `secpolicy * sp_in`
- `secpolicy * sp_out`
- `int priv`
- `secpolicy * cache [3]`
- `secpolicyindex cacheidx [3]`
- `int cachegen [3]`
- `int cacheflags`

6.42.1 Detailed Description

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

6.42.2 Field Documentation

6.42.2.1 `struct secpolicy* inpcbpolicy::cache[3]`

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

Referenced by `ipsec_checkpcbcache()`.

6.42.2.2 `int inpcbpolicy::cacheflags`

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

Referenced by `ipsec_checkpcbcache()`.

6.42.2.3 `int inpcbpolicy::cachegen[3]`

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

Referenced by `ipsec_checkpcbcache()`.

6.42.2.4 struct [secpolicyindex](#) [inpcbpolicy::cacheidx](#)[3]

Definition at line 134 of file [ipsec.h](#).

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

6.42.2.5 int [inpcbpolicy::priv](#)

Definition at line 129 of file [ipsec.h](#).

Referenced by [ipsec4_getpolicybypcb\(\)](#).

6.42.2.6 struct [secpolicy*](#) [inpcbpolicy::sp_in](#)

Definition at line 127 of file [ipsec.h](#).

Referenced by [ipsec4_getpolicybypcb\(\)](#).

6.42.2.7 struct [secpolicy*](#) [inpcbpolicy::sp_out](#)

Definition at line 128 of file [ipsec.h](#).

Referenced by [ipsec4_getpolicybypcb\(\)](#).

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.43 ip6_exthdrs Struct Reference

Data Fields

- mbuf * [ip6e_ip6](#)
- mbuf * [ip6e_hbh](#)
- mbuf * [ip6e_dest1](#)
- mbuf * [ip6e_rthdr](#)
- mbuf * [ip6e_dest2](#)

6.43.1 Detailed Description

Definition at line 113 of file ip6_output.c.

6.43.2 Field Documentation

6.43.2.1 struct mbuf* [ip6_exthdrs::ip6e_dest1](#)

Definition at line 116 of file ip6_output.c.

6.43.2.2 struct mbuf* [ip6_exthdrs::ip6e_dest2](#)

Definition at line 118 of file ip6_output.c.

6.43.2.3 struct mbuf* [ip6_exthdrs::ip6e_hbh](#)

Definition at line 115 of file ip6_output.c.

6.43.2.4 struct mbuf* [ip6_exthdrs::ip6e_ip6](#)

Definition at line 114 of file ip6_output.c.

Referenced by [ip6_splthdr\(\)](#).

6.43.2.5 struct mbuf* [ip6_exthdrs::ip6e_rthdr](#)

Definition at line 117 of file ip6_output.c.

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

- [/usr/src/sys/netinet6/ip6_output.c](#)

6.44 ip6_moptions Struct Reference

```
#include <ip6_var.h>
```

Public Member Functions

- [LIST_HEAD](#) ([in6_multi_mship](#)) [im6o_memberships](#)

Data Fields

- ifnet * [im6o_multicast_ifp](#)
- u_char [im6o_multicast_hlim](#)
- u_char [im6o_multicast_loop](#)

6.44.1 Detailed Description

Definition at line 108 of file [ip6_var.h](#).

6.44.2 Member Function Documentation

6.44.2.1 [ip6_moptions::LIST_HEAD \(in6_multi_mship\)](#)

6.44.3 Field Documentation

6.44.3.1 u_char [ip6_moptions::im6o_multicast_hlim](#)

Definition at line 110 of file [ip6_var.h](#).

Referenced by [ip6_getmoptions\(\)](#), [ip6_output\(\)](#), [ip6_setmoptions\(\)](#), [nd6_na_output\(\)](#), [nd6_ns_output\(\)](#), and [phyint_send\(\)](#).

6.44.3.2 struct ifnet* [ip6_moptions::im6o_multicast_ifp](#)

Definition at line 109 of file [ip6_var.h](#).

Referenced by [in6_pcbpurgeif0\(\)](#), [ip6_getmoptions\(\)](#), [ip6_setmoptions\(\)](#), [nd6_na_output\(\)](#), [nd6_ns_output\(\)](#), [phyint_send\(\)](#), and [selectroute\(\)](#).

6.44.3.3 u_char [ip6_moptions::im6o_multicast_loop](#)

Definition at line 111 of file [ip6_var.h](#).

Referenced by [ip6_getmoptions\(\)](#), [ip6_output\(\)](#), [ip6_setmoptions\(\)](#), [nd6_na_output\(\)](#), [nd6_ns_output\(\)](#), and [phyint_send\(\)](#).

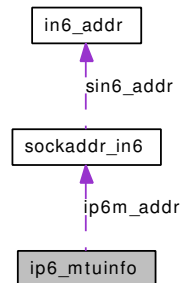
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.45 ip6_mtuinfo Struct Reference

```
#include <in6.h>
```

Collaboration diagram for ip6_mtuinfo:



Data Fields

- `sockaddr_in6 ip6m_addr`
- `uint32_t ip6m_mtu`

6.45.1 Detailed Description

Definition at line 501 of file `in6.h`.

6.45.2 Field Documentation

6.45.2.1 struct `sockaddr_in6 ip6_mtuinfo::ip6m_addr`

Definition at line 502 of file `in6.h`.

6.45.2.2 `uint32_t ip6_mtuinfo::ip6m_mtu`

Definition at line 503 of file `in6.h`.

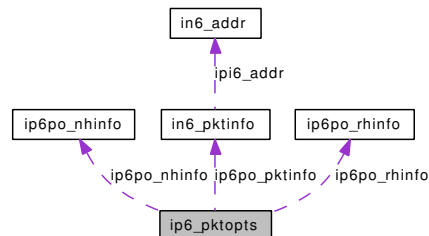
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6.h`

6.46 ip6_pktopts Struct Reference

```
#include <ip6_var.h>
```

Collaboration diagram for ip6_pktopts:



Data Fields

- mbuf * [ip6po_m](#)
- int [ip6po_hlim](#)
- [in6_pktinfo](#) * [ip6po_pktinfo](#)
- [ip6po_nhinfo](#) [ip6po_nhinfo](#)
- [ip6_hbh](#) * [ip6po_hbh](#)
- [ip6_dest](#) * [ip6po_dest1](#)
- [ip6po_rhinfo](#) [ip6po_rhinfo](#)
- [ip6_dest](#) * [ip6po_dest2](#)
- int [ip6po_tclass](#)
- int [ip6po_minmtu](#)
- int [ip6po_prefer_tempaddr](#)
- int [ip6po_flags](#)

6.46.1 Detailed Description

Definition at line 135 of file [ip6_var.h](#).

6.46.2 Field Documentation

6.46.2.1 struct ip6_dest* [ip6_pktopts::ip6po_dest1](#)

Definition at line 148 of file [ip6_var.h](#).

Referenced by [ip6_output\(\)](#), [ip6_pcbopts\(\)](#), and [ip6_setpktopt\(\)](#).

6.46.2.2 struct ip6_dest* [ip6_pktopts::ip6po_dest2](#)

Definition at line 154 of file [ip6_var.h](#).

Referenced by [ip6_output\(\)](#), [ip6_pcbopts\(\)](#), and [ip6_setpktopt\(\)](#).

6.46.2.3 int ip6_pktopts::ip6po_flags

Definition at line 169 of file ip6_var.h.

Referenced by ip6_output(), and ip6_setpktopt().

6.46.2.4 struct ip6_hbh* ip6_pktopts::ip6po_hbh

Definition at line 145 of file ip6_var.h.

Referenced by ip6_output(), ip6_pcbopts(), ip6_setpktopt(), and mld6_init().

6.46.2.5 int ip6_pktopts::ip6po_hlim

Definition at line 137 of file ip6_var.h.

Referenced by ip6_output(), and ip6_setpktopt().

6.46.2.6 struct mbuf* ip6_pktopts::ip6po_m

Definition at line 136 of file ip6_var.h.

6.46.2.7 int ip6_pktopts::ip6po_minmtu

Definition at line 158 of file ip6_var.h.

Referenced by ip6_output(), and ip6_setpktopt().

6.46.2.8 struct ip6po_nhinfo ip6_pktopts::ip6po_nhinfo

Definition at line 143 of file ip6_var.h.

6.46.2.9 struct in6_pktinfo* ip6_pktopts::ip6po_pktinfo

Definition at line 140 of file ip6_var.h.

Referenced by in6_selectsrc(), ip6_pcbopts(), ip6_setpktopt(), and selectroute().

6.46.2.10 int ip6_pktopts::ip6po_prefer_tempaddr

Definition at line 163 of file ip6_var.h.

Referenced by ip6_setpktopt().

6.46.2.11 struct ip6po_rhinfo ip6_pktopts::ip6po_rhinfo

Definition at line 151 of file ip6_var.h.

Referenced by ip6_pcbopts().

6.46.2.12 `int ip6_pktopts::ip6po_tclass`

Definition at line 156 of file `ip6_var.h`.

Referenced by `ip6_output()`, and `ip6_setpktopt()`.

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

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.47 ip6asfrag Struct Reference

```
#include <ip6_var.h>
```

Collaboration diagram for ip6asfrag:



Data Fields

- `u_int32_t ip6af_head`
- `u_int16_t ip6af_len`
- `u_int8_t ip6af_nxt`
- `u_int8_t ip6af_hlim`
- `ip6asfrag * ip6af_down`
- `ip6asfrag * ip6af_up`
- `mbuf * ip6af_m`
- `int ip6af_offset`
- `int ip6af_frglen`
- `int ip6af_off`
- `u_int16_t ip6af_mff`

6.47.1 Detailed Description

Definition at line 91 of file `ip6_var.h`.

6.47.2 Field Documentation

6.47.2.1 struct `ip6asfrag*` `ip6asfrag::ip6af_down`

Definition at line 97 of file `ip6_var.h`.

Referenced by `frag6_freef()`, and `frag6_input()`.

6.47.2.2 int `ip6asfrag::ip6af_frglen`

Definition at line 101 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.47.2.3 `u_int32_t` `ip6asfrag::ip6af_head`

Definition at line 92 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.47.2.4 `u_int8_t` `ip6asfrag::ip6af_hlim`

Definition at line 95 of file `ip6_var.h`.

6.47.2.5 u_int16_t ip6asfrag::ip6af_len

Definition at line 93 of file ip6_var.h.

6.47.2.6 struct mbuf* ip6asfrag::ip6af_m

Definition at line 99 of file ip6_var.h.

6.47.2.7 u_int16_t ip6asfrag::ip6af_mff

Definition at line 103 of file ip6_var.h.

Referenced by frag6_input().

6.47.2.8 u_int8_t ip6asfrag::ip6af_nxt

Definition at line 94 of file ip6_var.h.

6.47.2.9 int ip6asfrag::ip6af_off

Definition at line 102 of file ip6_var.h.

Referenced by frag6_freef(), and frag6_input().

6.47.2.10 int ip6asfrag::ip6af_offset

Definition at line 100 of file ip6_var.h.

Referenced by frag6_input().

6.47.2.11 struct ip6asfrag* ip6asfrag::ip6af_up

Definition at line 98 of file ip6_var.h.

Referenced by frag6_input().

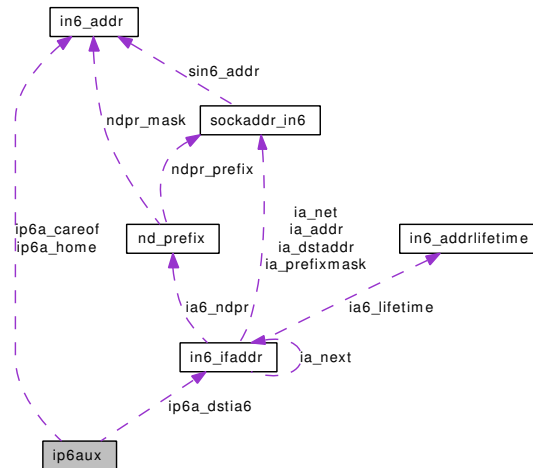
The documentation for this struct was generated from the following file:

- /usr/src/sys/netinet6/ip6_var.h

6.48 ip6aux Struct Reference

```
#include <ip6_var.h>
```

Collaboration diagram for ip6aux:



Data Fields

- `u_int32_t ip6a_flags`
- `in6_addr ip6a_careof`
- `in6_addr ip6a_home`
- `u_int16_t ip6a_bruid`
- `in6_ifaddr * ip6a_dstia6`
- `u_int16_t ip6a_rtalert`

6.48.1 Detailed Description

Definition at line 252 of file ip6_var.h.

6.48.2 Field Documentation

6.48.2.1 `u_int16_t ip6aux::ip6a_bruid`

Definition at line 262 of file ip6_var.h.

6.48.2.2 `struct in6_addr ip6aux::ip6a_careof`

Definition at line 260 of file ip6_var.h.

6.48.2.3 `struct in6_ifaddr* ip6aux::ip6a_dstia6`

Definition at line 265 of file ip6_var.h.

Referenced by `ip6_getdstifaddr()`, and `ip6_setdstifaddr()`.

6.48.2.4 u_int32_t ip6aux::ip6a_flags

Definition at line 253 of file ip6_var.h.

Referenced by route6_input().

6.48.2.5 struct in6_addr ip6aux::ip6a_home

Definition at line 261 of file ip6_var.h.

6.48.2.6 u_int16_t ip6aux::ip6a_rtalert

Definition at line 268 of file ip6_var.h.

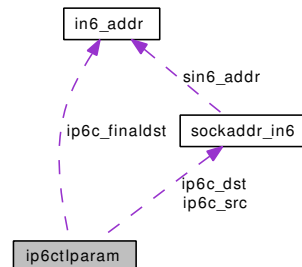
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.49 ip6ctlparam Struct Reference

```
#include <ip6protosw.h>
```

Collaboration diagram for ip6ctlparam:



Data Fields

- mbuf * [ip6c_m](#)
- icmp6_hdr * [ip6c_icmp6](#)
- ip6_hdr * [ip6c_ip6](#)
- int [ip6c_off](#)
- [sockaddr_in6](#) * [ip6c_src](#)
- [sockaddr_in6](#) * [ip6c_dst](#)
- [in6_addr](#) * [ip6c_finaldst](#)
- void * [ip6c_cmdarg](#)
- u_int8_t [ip6c_nxt](#)

6.49.1 Detailed Description

Definition at line 103 of file ip6protosw.h.

6.49.2 Field Documentation

6.49.2.1 void* [ip6ctlparam::ip6c_cmdarg](#)

Definition at line 111 of file ip6protosw.h.

Referenced by [icmp6_notify_error\(\)](#), [rip6_ctlinput\(\)](#), and [udp6_ctlinput\(\)](#).

6.49.2.2 struct [sockaddr_in6](#)* [ip6ctlparam::ip6c_dst](#)

Definition at line 109 of file ip6protosw.h.

6.49.2.3 struct [in6_addr](#)* [ip6ctlparam::ip6c_finaldst](#)

Definition at line 110 of file ip6protosw.h.

Referenced by [icmp6_notify_error\(\)](#).

6.49.2.4 struct icmp6_hdr* ip6ctlparam::ip6c_icmp6

Definition at line 105 of file ip6protosw.h.

Referenced by icmp6_notify_error().

6.49.2.5 struct ip6_hdr* ip6ctlparam::ip6c_ip6

Definition at line 106 of file ip6protosw.h.

Referenced by icmp6_notify_error(), rip6_ctlinput(), and udp6_ctlinput().

6.49.2.6 struct mbuf* ip6ctlparam::ip6c_m

Definition at line 104 of file ip6protosw.h.

Referenced by icmp6_notify_error(), rip6_ctlinput(), and udp6_ctlinput().

6.49.2.7 u_int8_t ip6ctlparam::ip6c_nxt

Definition at line 112 of file ip6protosw.h.

Referenced by icmp6_notify_error().

6.49.2.8 int ip6ctlparam::ip6c_off

Definition at line 107 of file ip6protosw.h.

Referenced by icmp6_notify_error(), rip6_ctlinput(), and udp6_ctlinput().

6.49.2.9 struct sockaddr_in6* ip6ctlparam::ip6c_src

Definition at line 108 of file ip6protosw.h.

Referenced by icmp6_notify_error(), rip6_ctlinput(), and udp6_ctlinput().

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

- [/usr/src/sys/netinet6/ip6protosw.h](#)

6.50 ip6po_nhinfo Struct Reference

```
#include <ip6_var.h>
```

Data Fields

- sockaddr * [ip6po_nhi_nexthop](#)
- route_in6 [ip6po_nhi_route](#)

6.50.1 Detailed Description

Definition at line 128 of file ip6_var.h.

6.50.2 Field Documentation

6.50.2.1 struct sockaddr* [ip6po_nhinfo::ip6po_nhi_nexthop](#)

Definition at line 129 of file ip6_var.h.

6.50.2.2 struct route_in6 [ip6po_nhinfo::ip6po_nhi_route](#)

Definition at line 130 of file ip6_var.h.

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

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.51 ip6po_rhinfo Struct Reference

```
#include <ip6_var.h>
```

Data Fields

- ip6_rthdr * [ip6po_rhi_rthdr](#)
- route_in6 [ip6po_rhi_route](#)

6.51.1 Detailed Description

Definition at line 120 of file ip6_var.h.

6.51.2 Field Documentation

6.51.2.1 struct route_in6 [ip6po_rhinfo::ip6po_rhi_route](#)

Definition at line 122 of file ip6_var.h.

6.51.2.2 struct ip6_rthdr* [ip6po_rhinfo::ip6po_rhi_rthdr](#)

Definition at line 121 of file ip6_var.h.

Referenced by [ip6_pcbopts\(\)](#).

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

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.52 ip6protosw Struct Reference

```
#include <ip6protosw.h>
```

Public Member Functions

- `int pr_input __P ((struct mbuf **, int *, int))`
- `int pr_output __P ((struct mbuf *,...))`
- `void pr_ctlinput __P ((int, struct sockaddr *, void *))`
- `int pr_ctloutput __P ((struct socket *, struct sockopt *))`
- `int pr_usrreq __P ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *, struct thread *))`
- `void pr_init __P ((void))`
- `void pr_fasttimo __P ((void))`
- `void pr_slowtimo __P ((void))`
- `void pr_drain __P ((void))`

Data Fields

- short `pr_type`
- domain * `pr_domain`
- short `pr_protocol`
- short `pr_flags`
- `pr_usrreqs * pr_usrreqs`

6.52.1 Detailed Description

Definition at line 115 of file ip6protosw.h.

6.52.2 Member Function Documentation

6.52.2.1 void `pr_drain ip6protosw::__P` ((void))

6.52.2.2 void `pr_slowtimo ip6protosw::__P` ((void))

6.52.2.3 void `pr_fasttimo ip6protosw::__P` ((void))

6.52.2.4 void `pr_init ip6protosw::__P` ((void))

6.52.2.5 int `pr_usrreq ip6protosw::__P` ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *, struct thread *))

6.52.2.6 int `pr_ctloutput ip6protosw::__P` ((struct socket *, struct sockopt *))

6.52.2.7 void `pr_ctlinput ip6protosw::__P` ((int, struct sockaddr *, void *))

6.52.2.8 int `pr_output ip6protosw::__P` ((struct mbuf *,...))

6.52.2.9 int `pr_input ip6protosw::__P` ((struct mbuf **, int *, int))

6.52.3 Field Documentation

6.52.3.1 struct domain* `ip6protosw::pr_domain`

Definition at line 117 of file `ip6protosw.h`.

Referenced by `ip6_init()`.

6.52.3.2 short `ip6protosw::pr_flags`

Definition at line 119 of file `ip6protosw.h`.

6.52.3.3 short `ip6protosw::pr_protocol`

Definition at line 118 of file `ip6protosw.h`.

Referenced by `ip6_init()`.

6.52.3.4 short `ip6protosw::pr_type`

Definition at line 116 of file `ip6protosw.h`.

6.52.3.5 struct `pr_usrreqs* ip6protosw::pr_usrreqs`

Definition at line 146 of file `ip6protosw.h`.

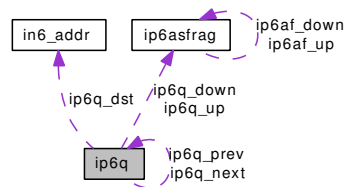
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/ip6protosw.h`

6.53 ip6q Struct Reference

```
#include <ip6_var.h>
```

Collaboration diagram for ip6q:



Data Fields

- `u_int32_t ip6q_head`
- `u_int16_t ip6q_len`
- `u_int8_t ip6q_nxt`
- `u_int8_t ip6q_hlim`
- `ip6asfrag * ip6q_down`
- `ip6asfrag * ip6q_up`
- `u_int32_t ip6q_ident`
- `u_int8_t ip6q_arrive`
- `u_int8_t ip6q_ttl`
- `in6_addr ip6q_src ip6q_dst`
- `ip6q * ip6q_next`
- `ip6q * ip6q_prev`
- `int ip6q_unfrglen`
- `int ip6q_nfrag`

6.53.1 Detailed Description

Definition at line 71 of file `ip6_var.h`.

6.53.2 Field Documentation

6.53.2.1 `u_int8_t ip6q::ip6q_arrive`

Definition at line 79 of file `ip6_var.h`.

6.53.2.2 `struct ip6asfrag* ip6q::ip6q_down`

Definition at line 76 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.53.2.3 struct `in6_addr ip6q_src ip6q::ip6q_dst`

Definition at line 81 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.53.2.4 u_int32_t `ip6q::ip6q_head`

Definition at line 72 of file `ip6_var.h`.

6.53.2.5 u_int8_t `ip6q::ip6q_hlim`

Definition at line 75 of file `ip6_var.h`.

6.53.2.6 u_int32_t `ip6q::ip6q_ident`

Definition at line 78 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.53.2.7 u_int16_t `ip6q::ip6q_len`

Definition at line 73 of file `ip6_var.h`.

6.53.2.8 struct `ip6q* ip6q::ip6q_next`

Definition at line 82 of file `ip6_var.h`.

Referenced by `frag6_drain()`, `frag6_init()`, `frag6_input()`, and `frag6_slowtimo()`.

6.53.2.9 int `ip6q::ip6q_nfrag`

Definition at line 88 of file `ip6_var.h`.

6.53.2.10 u_int8_t `ip6q::ip6q_nxt`

Definition at line 74 of file `ip6_var.h`.

Referenced by `frag6_input()`.

6.53.2.11 struct `ip6q* ip6q::ip6q_prev`

Definition at line 83 of file `ip6_var.h`.

Referenced by `frag6_init()`, and `frag6_slowtimo()`.

6.53.2.12 u_int8_t `ip6q::ip6q_ttl`

Definition at line 80 of file `ip6_var.h`.

Referenced by `frag6_slowtimo()`.

6.53.2.13 int [ip6q::ip6q_unfrglen](#)

Definition at line 84 of file [ip6_var.h](#).

Referenced by [frag6_input\(\)](#).

6.53.2.14 struct [ip6asfrag*](#) [ip6q::ip6q_up](#)

Definition at line 77 of file [ip6_var.h](#).

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

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.54 ip6stat Struct Reference

```
#include <ip6_var.h>
```

Data Fields

- u_quad_t ip6s_total
- u_quad_t ip6s_tooshort
- u_quad_t ip6s_toosmall
- u_quad_t ip6s_fragments
- u_quad_t ip6s_fragdropped
- u_quad_t ip6s_fragtimeout
- u_quad_t ip6s_fragoverflow
- u_quad_t ip6s_forward
- u_quad_t ip6s_cantforward
- u_quad_t ip6s_redirectsent
- u_quad_t ip6s_delivered
- u_quad_t ip6s_localout
- u_quad_t ip6s_odropped
- u_quad_t ip6s_reassembled
- u_quad_t ip6s_fragmented
- u_quad_t ip6s_ofragments
- u_quad_t ip6s_cantfrag
- u_quad_t ip6s_badoptions
- u_quad_t ip6s_noroute
- u_quad_t ip6s_badvers
- u_quad_t ip6s_rawout
- u_quad_t ip6s_badscope
- u_quad_t ip6s_notmember
- u_quad_t ip6s_nxthist [256]
- u_quad_t ip6s_m1
- u_quad_t ip6s_m2m [32]
- u_quad_t ip6s_mext1
- u_quad_t ip6s_mext2m
- u_quad_t ip6s_exthdrtoolong
- u_quad_t ip6s_nogif
- u_quad_t ip6s_toomanyhdr
- u_quad_t ip6s_sources_none
- u_quad_t ip6s_sources_sameif [16]
- u_quad_t ip6s_sources_otherif [16]
- u_quad_t ip6s_sources_samescope [16]
- u_quad_t ip6s_sources_otherscope [16]
- u_quad_t ip6s_sources_deprecated [16]
- u_quad_t ip6s_forward_cachehit
- u_quad_t ip6s_forward_cachemiss
- u_quad_t ip6s_sources_rule [16]

6.54.1 Detailed Description

Definition at line 182 of file ip6_var.h.

6.54.2 Field Documentation

6.54.2.1 `u_quad_t ip6stat::ip6s_badoptions`

Definition at line 200 of file ip6_var.h.

Referenced by ip6_input(), ip6_process_hopopts(), and ip6_unknown_opt().

6.54.2.2 `u_quad_t ip6stat::ip6s_badscope`

Definition at line 204 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.3 `u_quad_t ip6stat::ip6s_badvers`

Definition at line 202 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.4 `u_quad_t ip6stat::ip6s_cantforward`

Definition at line 191 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.5 `u_quad_t ip6stat::ip6s_cantfrag`

Definition at line 199 of file ip6_var.h.

6.54.2.6 `u_quad_t ip6stat::ip6s_delivered`

Definition at line 193 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.7 `u_quad_t ip6stat::ip6s_exthdrtoolong`

Definition at line 211 of file ip6_var.h.

6.54.2.8 `u_quad_t ip6stat::ip6s_forward`

Definition at line 190 of file ip6_var.h.

6.54.2.9 `u_quad_t ip6stat::ip6s_forward_cachehit`

Definition at line 239 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.10 u_quad_t ip6stat::ip6s_forward_cachemiss

Definition at line 240 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.11 u_quad_t ip6stat::ip6s_fragdropped

Definition at line 187 of file ip6_var.h.

6.54.2.12 u_quad_t ip6stat::ip6s_fragmented

Definition at line 197 of file ip6_var.h.

6.54.2.13 u_quad_t ip6stat::ip6s_fragments

Definition at line 186 of file ip6_var.h.

6.54.2.14 u_quad_t ip6stat::ip6s_fragoverflow

Definition at line 189 of file ip6_var.h.

6.54.2.15 u_quad_t ip6stat::ip6s_fragtimeout

Definition at line 188 of file ip6_var.h.

6.54.2.16 u_quad_t ip6stat::ip6s_localout

Definition at line 194 of file ip6_var.h.

6.54.2.17 u_quad_t ip6stat::ip6s_m1

Definition at line 207 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.18 u_quad_t ip6stat::ip6s_m2m[32]

Definition at line 208 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.19 u_quad_t ip6stat::ip6s_mext1

Definition at line 209 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.20 u_quad_t ip6stat::ip6s_mext2m

Definition at line 210 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.21 u_quad_t ip6stat::ip6s_nogif

Definition at line 212 of file ip6_var.h.

6.54.2.22 u_quad_t ip6stat::ip6s_noroute

Definition at line 201 of file ip6_var.h.

6.54.2.23 u_quad_t ip6stat::ip6s_notmember

Definition at line 205 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.24 u_quad_t ip6stat::ip6s_nxthist[256]

Definition at line 206 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.25 u_quad_t ip6stat::ip6s_odropped

Definition at line 195 of file ip6_var.h.

6.54.2.26 u_quad_t ip6stat::ip6s_ofragments

Definition at line 198 of file ip6_var.h.

6.54.2.27 u_quad_t ip6stat::ip6s_rawout

Definition at line 203 of file ip6_var.h.

6.54.2.28 u_quad_t ip6stat::ip6s_reassembled

Definition at line 196 of file ip6_var.h.

6.54.2.29 u_quad_t ip6stat::ip6s_redirectsent

Definition at line 192 of file ip6_var.h.

6.54.2.30 u_quad_t ip6stat::ip6s_sources_deprecated[16]

Definition at line 237 of file ip6_var.h.

6.54.2.31 u_quad_t ip6stat::ip6s_sources_none

Definition at line 221 of file ip6_var.h.

6.54.2.32 u_quad_t ip6stat::ip6s_sources_otherif[16]

Definition at line 225 of file ip6_var.h.

6.54.2.33 u_quad_t ip6stat::ip6s_sources_otherscope[16]

Definition at line 235 of file ip6_var.h.

6.54.2.34 u_quad_t ip6stat::ip6s_sources_rule[16]

Definition at line 243 of file ip6_var.h.

6.54.2.35 u_quad_t ip6stat::ip6s_sources_sameif[16]

Definition at line 223 of file ip6_var.h.

6.54.2.36 u_quad_t ip6stat::ip6s_sources_samescope[16]

Definition at line 230 of file ip6_var.h.

6.54.2.37 u_quad_t ip6stat::ip6s_toomanyhdr

Definition at line 213 of file ip6_var.h.

Referenced by ip6_input().

6.54.2.38 u_quad_t ip6stat::ip6s_tooshort

Definition at line 184 of file ip6_var.h.

Referenced by ip6_hopopts_input(), ip6_input(), and ip6_savecontrol().

6.54.2.39 u_quad_t ip6stat::ip6s_toosmall

Definition at line 185 of file ip6_var.h.

Referenced by ip6_input(), and ip6_process_hopopts().

6.54.2.40 u_quad_t ip6stat::ip6s_total

Definition at line 183 of file ip6_var.h.

Referenced by ip6_input().

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

- [/usr/src/sys/netinet6/ip6_var.h](#)

6.55 ipcomp Struct Reference

```
#include <ipcomp.h>
```

Data Fields

- [u_int8_t comp_nxt](#)
- [u_int8_t comp_flags](#)
- [u_int16_t comp_cpi](#)

6.55.1 Detailed Description

Definition at line 44 of file ipcomp.h.

6.55.2 Field Documentation

6.55.2.1 [u_int16_t ipcomp::comp_cpi](#)

Definition at line 47 of file ipcomp.h.

6.55.2.2 [u_int8_t ipcomp::comp_flags](#)

Definition at line 46 of file ipcomp.h.

6.55.2.3 [u_int8_t ipcomp::comp_nxt](#)

Definition at line 45 of file ipcomp.h.

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

- [/usr/src/sys/netinet6/ipcomp.h](#)

6.56 ipcomp_algorithm Struct Reference

```
#include <ipcomp.h>
```

Public Member Functions

- int compress [__P](#) ((struct mbuf *, struct mbuf *, [size_t](#) *)
- int decompress [__P](#) ((struct mbuf *, struct mbuf *, [size_t](#) *)

Data Fields

- [size_t minplen](#)

6.56.1 Detailed Description

Definition at line 59 of file ipcomp.h.

6.56.2 Member Function Documentation

6.56.2.1 int decompress ipcomp_algorithm::__P ((struct mbuf *, struct mbuf *, [size_t](#) *)

6.56.2.2 int compress ipcomp_algorithm::__P ((struct mbuf *, struct mbuf *, [size_t](#) *)

6.56.3 Field Documentation

6.56.3.1 [size_t ipcomp_algorithm::minplen](#)

Definition at line 62 of file ipcomp.h.

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

- [/usr/src/sys/netinet6/ipcomp.h](#)

6.57 ipsec_history Struct Reference

```
#include <ipsec.h>
```

Data Fields

- int [ih_proto](#)
- u_int32_t [ih_spi](#)

6.57.1 Detailed Description

Definition at line 309 of file ipsec.h.

6.57.2 Field Documentation

6.57.2.1 int [ipsec_history::ih_proto](#)

Definition at line 310 of file ipsec.h.

6.57.2.2 u_int32_t [ipsec_history::ih_spi](#)

Definition at line 311 of file ipsec.h.

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.58 ipsec_output_state Struct Reference

```
#include <ipsec.h>
```

Data Fields

- mbuf * [m](#)
- route * [ro](#)
- sockaddr * [dst](#)
- int [encap](#)

6.58.1 Detailed Description

Definition at line 302 of file ipsec.h.

6.58.2 Field Documentation

6.58.2.1 struct sockaddr* [ipsec_output_state::dst](#)

Definition at line 305 of file ipsec.h.

Referenced by [ip6_forward\(\)](#).

6.58.2.2 int [ipsec_output_state::encap](#)

Definition at line 306 of file ipsec.h.

6.58.2.3 struct mbuf* [ipsec_output_state::m](#)

Definition at line 303 of file ipsec.h.

6.58.2.4 struct route* [ipsec_output_state::ro](#)

Definition at line 304 of file ipsec.h.

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.59 ipsecaux Struct Reference

```
#include <ipsec.h>
```

Data Fields

- int [hdrs](#)

6.59.1 Detailed Description

Definition at line 151 of file ipsec.h.

6.59.2 Field Documentation

6.59.2.1 int [ipsecaux::hdrs](#)

Definition at line 152 of file ipsec.h.

Referenced by ipsec_addhist(), and ipsec_getnhist().

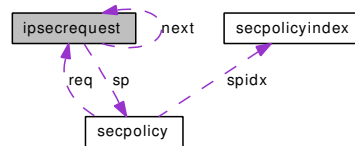
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ipsec.h](#)

6.60 ipsecrequest Struct Reference

```
#include <ipsec.h>
```

Collaboration diagram for ipsecrequest:



Data Fields

- [ipsecrequest * next](#)
- [secasindex saidx](#)
- [u_int level](#)
- [secasvar * sav](#)
- [secpolicy * sp](#)
- [ifnet * tunifp](#)

6.60.1 Detailed Description

Definition at line 111 of file ipsec.h.

6.60.2 Field Documentation

6.60.2.1 [u_int ipsecrequest::level](#)

Definition at line 117 of file ipsec.h.

Referenced by [ipsec_deepcopy_policy\(\)](#), and [ipsec_in_reject\(\)](#).

6.60.2.2 [struct ipsecrequest* ipsecrequest::next](#)

Definition at line 112 of file ipsec.h.

Referenced by [ip6_forward\(\)](#), [ipsec_deepcopy_policy\(\)](#), [ipsec_hdrsiz\(\)](#), and [ipsec_in_reject\(\)](#).

6.60.2.3 [struct secasindex ipsecrequest::saidx](#)

Definition at line 115 of file ipsec.h.

Referenced by [ip6_forward\(\)](#), [ipsec_deepcopy_policy\(\)](#), [ipsec_hdrsiz\(\)](#), and [ipsec_in_reject\(\)](#).

6.60.2.4 [struct secasvar* ipsecrequest::sav](#)

Definition at line 119 of file ipsec.h.

Referenced by [esp_output\(\)](#), [ipcomp_output\(\)](#), and [ipsec_in_reject\(\)](#).

6.60.2.5 struct [secpolicy*](#) [ipsecrequest::sp](#)

Definition at line 120 of file ipsec.h.

6.60.2.6 struct [ifnet*](#) [ipsecrequest::tunifp](#)

Definition at line 122 of file ipsec.h.

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.61 ipsecstat Struct Reference

```
#include <ipsec.h>
```

Data Fields

- [u_quad_t in_success](#)
- [u_quad_t in_polvio](#)
- [u_quad_t in_nosa](#)
- [u_quad_t in_inval](#)
- [u_quad_t in_nomem](#)
- [u_quad_t in_badspi](#)
- [u_quad_t in_ahreplay](#)
- [u_quad_t in_espreplay](#)
- [u_quad_t in_ahauthsucc](#)
- [u_quad_t in_ahauthfail](#)
- [u_quad_t in_espauthsucc](#)
- [u_quad_t in_espauthfail](#)
- [u_quad_t in_esphist](#) [256]
- [u_quad_t in_ahhist](#) [256]
- [u_quad_t in_comphist](#) [256]
- [u_quad_t out_success](#)
- [u_quad_t out_polvio](#)
- [u_quad_t out_nosa](#)
- [u_quad_t out_inval](#)
- [u_quad_t out_nomem](#)
- [u_quad_t out_noroute](#)
- [u_quad_t out_esphist](#) [256]
- [u_quad_t out_ahhist](#) [256]
- [u_quad_t out_comphist](#) [256]
- [u_quad_t spdcahelookup](#)
- [u_quad_t spdcahemiss](#)

6.61.1 Detailed Description

Definition at line 212 of file ipsec.h.

6.61.2 Field Documentation

6.61.2.1 [u_quad_t ipsecstat::in_ahauthfail](#)

Definition at line 223 of file ipsec.h.

6.61.2.2 [u_quad_t ipsecstat::in_ahauthsucc](#)

Definition at line 222 of file ipsec.h.

6.61.2.3 u_quad_t ipsecstat::in_ahhist[256]

Definition at line 227 of file ipsec.h.

6.61.2.4 u_quad_t ipsecstat::in_ahreplay

Definition at line 220 of file ipsec.h.

6.61.2.5 u_quad_t ipsecstat::in_badspi

Definition at line 219 of file ipsec.h.

6.61.2.6 u_quad_t ipsecstat::in_comphist[256]

Definition at line 228 of file ipsec.h.

6.61.2.7 u_quad_t ipsecstat::in_espauthfail

Definition at line 225 of file ipsec.h.

6.61.2.8 u_quad_t ipsecstat::in_espauthsucc

Definition at line 224 of file ipsec.h.

6.61.2.9 u_quad_t ipsecstat::in_espghost[256]

Definition at line 226 of file ipsec.h.

6.61.2.10 u_quad_t ipsecstat::in_espghost

Definition at line 221 of file ipsec.h.

6.61.2.11 u_quad_t ipsecstat::in_inval

Definition at line 217 of file ipsec.h.

6.61.2.12 u_quad_t ipsecstat::in_nomem

Definition at line 218 of file ipsec.h.

6.61.2.13 u_quad_t ipsecstat::in_nosa

Definition at line 216 of file ipsec.h.

6.61.2.14 u_quad_t ipsecstat::in_polvio

Definition at line 214 of file ipsec.h.

Referenced by ip6_forward(), ip6_input(), rip6_input(), sctp6_input(), and udp6_append().

6.61.2.15 u_quad_t ipsecstat::in_success

Definition at line 213 of file ipsec.h.

6.61.2.16 u_quad_t ipsecstat::out_ahhist[256]

Definition at line 237 of file ipsec.h.

6.61.2.17 u_quad_t ipsecstat::out_comphist[256]

Definition at line 238 of file ipsec.h.

6.61.2.18 u_quad_t ipsecstat::out_esphist[256]

Definition at line 236 of file ipsec.h.

6.61.2.19 u_quad_t ipsecstat::out_inval

Definition at line 233 of file ipsec.h.

Referenced by esp_output(), ip6_forward(), ip6_output(), and ipcomp_output().

6.61.2.20 u_quad_t ipsecstat::out_nomem

Definition at line 234 of file ipsec.h.

6.61.2.21 u_quad_t ipsecstat::out_noroute

Definition at line 235 of file ipsec.h.

6.61.2.22 u_quad_t ipsecstat::out_nosa

Definition at line 232 of file ipsec.h.

6.61.2.23 u_quad_t ipsecstat::out_polvio

Definition at line 230 of file ipsec.h.

Referenced by ip6_forward(), and ip6_output().

6.61.2.24 `u_quad_t ipsecstat::out_success`

Definition at line 229 of file ipsec.h.

6.61.2.25 `u_quad_t ipsecstat::spdcachelookup`

Definition at line 240 of file ipsec.h.

Referenced by ipsec4_getpolicybypcb().

6.61.2.26 `u_quad_t ipsecstat::spdcachemiss`

Definition at line 241 of file ipsec.h.

Referenced by ipsec4_getpolicybypcb().

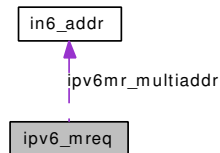
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ipsec.h](#)

6.62 `ipv6_mreq` Struct Reference

```
#include <in6.h>
```

Collaboration diagram for `ipv6_mreq`:



Data Fields

- `in6_addr` `ipv6mr_multiaddr`
- unsigned int `ipv6mr_interface`

6.62.1 Detailed Description

Definition at line 485 of file `in6.h`.

6.62.2 Field Documentation

6.62.2.1 unsigned int `ipv6_mreq::ipv6mr_interface`

Definition at line 487 of file `in6.h`.

Referenced by `ip6_setmoptions()`.

6.62.2.2 struct `in6_addr` `ipv6_mreq::ipv6mr_multiaddr`

Definition at line 486 of file `in6.h`.

Referenced by `ip6_setmoptions()`.

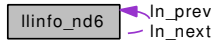
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/in6.h`

6.63 linfo_nd6 Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for linfo_nd6:



Data Fields

- `linfo_nd6 * ln_next`
- `linfo_nd6 * ln_prev`
- `rtentry * ln_rt`
- `mbuf * ln_hold`
- `long ln_asked`
- `u_long ln_expire`
- `short ln_state`
- `short ln_router`
- `int ln_byhint`
- `long ln_ntick`
- `callout ln_timer_ch`

6.63.1 Detailed Description

Definition at line 44 of file nd6.h.

6.63.2 Field Documentation

6.63.2.1 `long linfo_nd6::ln_asked`

Definition at line 49 of file nd6.h.

Referenced by `nd6_ioctl()`, `nd6_linfo_timer()`, and `nd6_na_input()`.

6.63.2.2 `int linfo_nd6::ln_byhint`

Definition at line 53 of file nd6.h.

Referenced by `nd6_na_input()`, and `nd6_nud_hint()`.

6.63.2.3 `u_long linfo_nd6::ln_expire`

Definition at line 50 of file nd6.h.

Referenced by `nd6_ioctl()`, `nd6_linfo_settimer()`, and `nd6_linfo_timer()`.

6.63.2.4 struct mbuf* linfo_nd6::ln_hold

Definition at line 48 of file nd6.h.

Referenced by nd6_llinfo_timer(), nd6_na_input(), and nd6_ns_output().

6.63.2.5 struct linfo_nd6* linfo_nd6::ln_next

Definition at line 45 of file nd6.h.

Referenced by nd6_free().

6.63.2.6 long linfo_nd6::ln_ntick

Definition at line 55 of file nd6.h.

Referenced by nd6_llinfo_timer().

6.63.2.7 struct linfo_nd6* linfo_nd6::ln_prev

Definition at line 46 of file nd6.h.

6.63.2.8 short linfo_nd6::ln_router

Definition at line 52 of file nd6.h.

Referenced by nd6_cache_lladdr(), nd6_free(), nd6_ioctl(), and nd6_na_input().

6.63.2.9 struct rtbody* linfo_nd6::ln_rt

Definition at line 47 of file nd6.h.

Referenced by nd6_llinfo_timer().

6.63.2.10 short linfo_nd6::ln_state

Definition at line 51 of file nd6.h.

Referenced by in6_ifinit(), nd6_cache_lladdr(), nd6_free(), nd6_ioctl(), nd6_llinfo_timer(), nd6_lookup(), nd6_na_input(), and nd6_nud_hint().

6.63.2.11 struct callout linfo_nd6::ln_timer_ch

Definition at line 56 of file nd6.h.

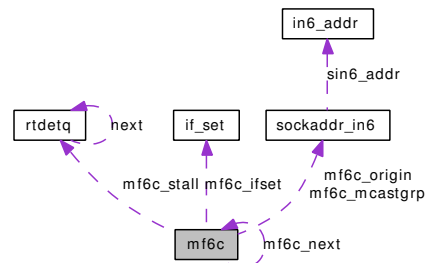
The documentation for this struct was generated from the following file:

- /usr/src/sys/netinet6/nd6.h

6.64 mf6c Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for mf6c:



Data Fields

- `sockaddr_in6` `mf6c_origin`
- `sockaddr_in6` `mf6c_mcastgrp`
- `mifi_t` `mf6c_parent`
- `if_set` `mf6c_ifset`
- `u_quad_t` `mf6c_pkt_cnt`
- `u_quad_t` `mf6c_byte_cnt`
- `u_quad_t` `mf6c_wrong_if`
- `int` `mf6c_expire`
- `timeval` `mf6c_last_assert`
- `rtdetq` * `mf6c_stall`
- `mf6c` * `mf6c_next`

6.64.1 Detailed Description

Definition at line 231 of file `ip6_mroute.h`.

6.64.2 Field Documentation

6.64.2.1 `u_quad_t` `mf6c::mf6c_byte_cnt`

Definition at line 238 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `get_sg_cnt()`, and `ip6_mdq()`.

6.64.2.2 `int` `mf6c::mf6c_expire`

Definition at line 240 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, and `expire_upcalls()`.

6.64.2.3 struct [if_set](#) `mf6c::mf6c_ifset`

Definition at line 235 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, and `ip6_mdq()`.

6.64.2.4 struct [timeval](#) `mf6c::mf6c_last_assert`

Definition at line 241 of file `ip6_mroute.h`.

6.64.2.5 struct [sockaddr_in6](#) `mf6c::mf6c_mcastgrp`

Definition at line 233 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, and `ip6_mforward()`.

6.64.2.6 struct [mf6c*](#) `mf6c::mf6c_next`

Definition at line 243 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, and `ip6_mforward()`.

6.64.2.7 struct [sockaddr_in6](#) `mf6c::mf6c_origin`

Definition at line 232 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, and `ip6_mforward()`.

6.64.2.8 [mifi_t](#) `mf6c::mf6c_parent`

Definition at line 234 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, and `ip6_mdq()`.

6.64.2.9 [u_quad_t](#) `mf6c::mf6c_pkt_cnt`

Definition at line 237 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `get_sg_cnt()`, and `ip6_mdq()`.

6.64.2.10 struct [rtdetq*](#) `mf6c::mf6c_stall`

Definition at line 242 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, and `ip6_mforward()`.

6.64.2.11 [u_quad_t](#) `mf6c::mf6c_wrong_if`

Definition at line 239 of file `ip6_mroute.h`.

Referenced by `add_m6fc()`, `get_sg_cnt()`, and `ip6_mdq()`.

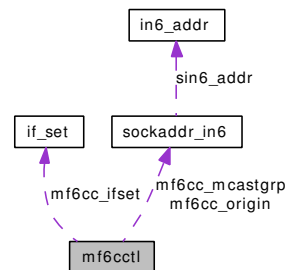
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.65 mf6ctl Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for mf6ctl:



Data Fields

- [sockaddr_in6 mf6cc_origin](#)
- [sockaddr_in6 mf6cc_mcastgrp](#)
- [mifi_t mf6cc_parent](#)
- [if_set mf6cc_ifset](#)

6.65.1 Detailed Description

Definition at line 116 of file ip6_mroute.h.

6.65.2 Field Documentation

6.65.2.1 struct [if_set](#) [mf6ctl::mf6cc_ifset](#)

Definition at line 120 of file ip6_mroute.h.

6.65.2.2 struct [sockaddr_in6](#) [mf6ctl::mf6cc_mcastgrp](#)

Definition at line 118 of file ip6_mroute.h.

6.65.2.3 struct [sockaddr_in6](#) [mf6ctl::mf6cc_origin](#)

Definition at line 117 of file ip6_mroute.h.

6.65.2.4 [mifi_t](#) [mf6ctl::mf6cc_parent](#)

Definition at line 119 of file ip6_mroute.h.

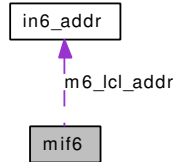
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.66 mif6 Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for mif6:



Data Fields

- u_char [m6_flags](#)
- u_int [m6_rate_limit](#)
- in6_addr [m6_lcl_addr](#)
- ifnet * [m6_ifp](#)
- u_quad_t [m6_pkt_in](#)
- u_quad_t [m6_pkt_out](#)
- u_quad_t [m6_bytes_in](#)
- u_quad_t [m6_bytes_out](#)
- route_in6 [m6_route](#)

6.66.1 Detailed Description

Definition at line 209 of file ip6_mroute.h.

6.66.2 Field Documentation

6.66.2.1 u_quad_t mif6::m6_bytes_in

Definition at line 219 of file ip6_mroute.h.

Referenced by [add_m6if\(\)](#), [get_mif6_cnt\(\)](#), and [ip6_mdq\(\)](#).

6.66.2.2 u_quad_t mif6::m6_bytes_out

Definition at line 220 of file ip6_mroute.h.

Referenced by [add_m6if\(\)](#), [get_mif6_cnt\(\)](#), and [ip6_mdq\(\)](#).

6.66.2.3 u_char mif6::m6_flags

Definition at line 210 of file ip6_mroute.h.

Referenced by [add_m6if\(\)](#), [del_m6if\(\)](#), and [ip6_mdq\(\)](#).

6.66.2.4 struct ifnet* mif6::m6_ifp

Definition at line 216 of file ip6_mroute.h.

Referenced by add_m6if(), del_m6if(), ip6_mdq(), ip6_mforward(), and phyint_send().

6.66.2.5 struct in6_addr mif6::m6_lcl_addr

Definition at line 215 of file ip6_mroute.h.

6.66.2.6 u_quad_t mif6::m6_pkt_in

Definition at line 217 of file ip6_mroute.h.

Referenced by add_m6if(), get_mif6_cnt(), and ip6_mdq().

6.66.2.7 u_quad_t mif6::m6_pkt_out

Definition at line 218 of file ip6_mroute.h.

Referenced by add_m6if(), get_mif6_cnt(), and ip6_mdq().

6.66.2.8 u_int mif6::m6_rate_limit

Definition at line 211 of file ip6_mroute.h.

Referenced by add_m6if().

6.66.2.9 struct route_in6 mif6::m6_route

Definition at line 221 of file ip6_mroute.h.

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

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.67 mif6ctl Struct Reference

```
#include <ip6_mroute.h>
```

Data Fields

- [mifi_t mif6c_mifi](#)
- [u_char mif6c_flags](#)
- [u_short mif6c_pifi](#)

6.67.1 Detailed Description

Definition at line 102 of file `ip6_mroute.h`.

6.67.2 Field Documentation

6.67.2.1 [u_char mif6ctl::mif6c_flags](#)

Definition at line 104 of file `ip6_mroute.h`.

6.67.2.2 [mifi_t mif6ctl::mif6c_mifi](#)

Definition at line 103 of file `ip6_mroute.h`.

6.67.2.3 [u_short mif6ctl::mif6c_pifi](#)

Definition at line 105 of file `ip6_mroute.h`.

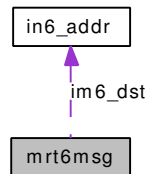
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.68 mrt6msg Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for mrt6msg:



Data Fields

- u_char [im6_mbz](#)
- u_char [im6_msgtype](#)
- u_int16_t [im6_mif](#)
- u_int32_t [im6_pad](#)
- [in6_addr](#) [im6_src](#) [im6_dst](#)

6.68.1 Detailed Description

Definition at line 171 of file ip6_mroute.h.

6.68.2 Field Documentation

6.68.2.1 struct [in6_addr](#) [im6_src](#) [mrt6msg::im6_dst](#)

Definition at line 179 of file ip6_mroute.h.

6.68.2.2 u_char [mrt6msg::im6_mbz](#)

Definition at line 175 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [register_send\(\)](#).

6.68.2.3 u_int16_t [mrt6msg::im6_mif](#)

Definition at line 177 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [register_send\(\)](#).

6.68.2.4 u_char [mrt6msg::im6_msgtype](#)

Definition at line 176 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [register_send\(\)](#).

6.68.2.5 `u_int32_t mrt6msg::im6_pad`

Definition at line 178 of file `ip6_mroute.h`.

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

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.69 mrt6stat Struct Reference

```
#include <ip6_mroute.h>
```

Data Fields

- [u_quad_t mrt6s_mfc_lookups](#)
- [u_quad_t mrt6s_mfc_misses](#)
- [u_quad_t mrt6s_upcalls](#)
- [u_quad_t mrt6s_no_route](#)
- [u_quad_t mrt6s_bad_tunnel](#)
- [u_quad_t mrt6s_cant_tunnel](#)
- [u_quad_t mrt6s_wrong_if](#)
- [u_quad_t mrt6s_upq_ovflw](#)
- [u_quad_t mrt6s_cache_cleanups](#)
- [u_quad_t mrt6s_drop_sel](#)
- [u_quad_t mrt6s_q_overflow](#)
- [u_quad_t mrt6s_pkt2large](#)
- [u_quad_t mrt6s_upq_sockfull](#)

6.69.1 Detailed Description

Definition at line 126 of file ip6_mroute.h.

6.69.2 Field Documentation

6.69.2.1 [u_quad_t mrt6stat::mrt6s_bad_tunnel](#)

Definition at line 131 of file ip6_mroute.h.

6.69.2.2 [u_quad_t mrt6stat::mrt6s_cache_cleanups](#)

Definition at line 135 of file ip6_mroute.h.

Referenced by `expire_upcalls()`.

6.69.2.3 [u_quad_t mrt6stat::mrt6s_cant_tunnel](#)

Definition at line 132 of file ip6_mroute.h.

6.69.2.4 [u_quad_t mrt6stat::mrt6s_drop_sel](#)

Definition at line 136 of file ip6_mroute.h.

6.69.2.5 [u_quad_t mrt6stat::mrt6s_mfc_lookups](#)

Definition at line 127 of file ip6_mroute.h.

6.69.2.6 `u_quad_t mrt6stat::mrt6s_mfc_misses`

Definition at line 128 of file `ip6_mroute.h`.

6.69.2.7 `u_quad_t mrt6stat::mrt6s_no_route`

Definition at line 130 of file `ip6_mroute.h`.

Referenced by `ip6_mforward()`.

6.69.2.8 `u_quad_t mrt6stat::mrt6s_pkt2large`

Definition at line 138 of file `ip6_mroute.h`.

6.69.2.9 `u_quad_t mrt6stat::mrt6s_q_overflow`

Definition at line 137 of file `ip6_mroute.h`.

6.69.2.10 `u_quad_t mrt6stat::mrt6s_upcalls`

Definition at line 129 of file `ip6_mroute.h`.

Referenced by `ip6_mdq()`, `ip6_mforward()`, and `register_send()`.

6.69.2.11 `u_quad_t mrt6stat::mrt6s_upq_ovflw`

Definition at line 134 of file `ip6_mroute.h`.

Referenced by `ip6_mforward()`.

6.69.2.12 `u_quad_t mrt6stat::mrt6s_upq_sockfull`

Definition at line 139 of file `ip6_mroute.h`.

Referenced by `ip6_mdq()`, `ip6_mforward()`, and `register_send()`.

6.69.2.13 `u_quad_t mrt6stat::mrt6s_wrong_if`

Definition at line 133 of file `ip6_mroute.h`.

Referenced by `ip6_mdq()`.

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

- `/usr/src/sys/netinet6/ip6_mroute.h`

6.70 mtuex_arg Struct Reference

Data Fields

- radix_node_head * [rnh](#)
- time_t [nextstop](#)

6.70.1 Detailed Description

Definition at line 377 of file [in6_rmx.c](#).

6.70.2 Field Documentation

6.70.2.1 time_t [mtuex_arg::nextstop](#)

Definition at line 379 of file [in6_rmx.c](#).

Referenced by [in6_mtuexpire\(\)](#), and [in6_mtutimo\(\)](#).

6.70.2.2 struct radix_node_head* [mtuex_arg::rnh](#)

Definition at line 378 of file [in6_rmx.c](#).

Referenced by [in6_mtutimo\(\)](#).

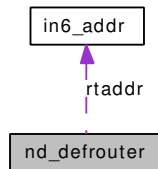
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/in6_rmx.c](#)

6.71 nd_defrouter Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for nd_defrouter:



Public Member Functions

- [TAILQ_ENTRY \(nd_defrouter\)](#) dr_entry

Data Fields

- [in6_addr](#) rtaddr
- [u_char](#) flags
- [u_short](#) rtlifetime
- [u_long](#) expire
- [ifnet *](#) ifp
- [int](#) installed

6.71.1 Detailed Description

Definition at line 244 of file nd6.h.

6.71.2 Member Function Documentation

6.71.2.1 nd_defrouter::TAILQ_ENTRY (nd_defrouter)

6.71.3 Field Documentation

6.71.3.1 u_long nd_defrouter::expire

Definition at line 249 of file nd6.h.

Referenced by defrtrlist_update(), nd6_free(), nd6_ioctl(), nd6_sysctl_drlist(), and nd6_timer().

6.71.3.2 u_char nd_defrouter::flags

Definition at line 247 of file nd6.h.

Referenced by defrtrlist_update(), nd6_ioctl(), nd6_sysctl_drlist(), and rtpref().

6.71.3.3 struct ifnet* nd_defrouter::ifp

Definition at line 250 of file nd6.h.

Referenced by defrouter_lookup(), defrouter_select(), find_pfxlist_reachable_router(), nd6_ioctl(), nd6_purge(), and nd6_sysctl_drlist().

6.71.3.4 int nd_defrouter::installed

Definition at line 251 of file nd6.h.

Referenced by defrouter_select(), and nd6_purge().

6.71.3.5 struct in6_addr nd_defrouter::rtaddr

Definition at line 246 of file nd6.h.

Referenced by defrouter_lookup(), defrouter_select(), find_pfxlist_reachable_router(), nd6_ioctl(), nd6_sysctl_drlist(), and nd6_sysctl_prlist().

6.71.3.6 u_short nd_defrouter::rtlifetime

Definition at line 248 of file nd6.h.

Referenced by defrtrlist_update(), nd6_ioctl(), and nd6_sysctl_drlist().

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

- /usr/src/sys/netinet6/nd6.h

6.72 nd_ifinfo Struct Reference

```
#include <nd6.h>
```

Data Fields

- `u_int32_t linkmtu`
- `u_int32_t maxmtu`
- `u_int32_t basereachable`
- `u_int32_t reachable`
- `u_int32_t retrans`
- `u_int32_t flags`
- `int recalctm`
- `u_int8_t chlim`
- `u_int8_t initialized`
- `u_int8_t randomseed0` [8]
- `u_int8_t randomseed1` [8]
- `u_int8_t randomid` [8]

6.72.1 Detailed Description

Definition at line 77 of file nd6.h.

6.72.2 Field Documentation

6.72.2.1 `u_int32_t nd_ifinfo::basereachable`

Definition at line 80 of file nd6.h.

Referenced by `nd6_ra_input()`, and `nd6_slowtimo()`.

6.72.2.2 `u_int8_t nd_ifinfo::chlim`

Definition at line 85 of file nd6.h.

Referenced by `nd6_ra_input()`.

6.72.2.3 `u_int32_t nd_ifinfo::flags`

Definition at line 83 of file nd6.h.

Referenced by `nd6_llinfo_timer()`, and `nd6_ra_input()`.

6.72.2.4 `u_int8_t nd_ifinfo::initialized`

Definition at line 86 of file nd6.h.

6.72.2.5 u_int32_t nd_ifinfo::linkmtu

Definition at line 78 of file nd6.h.

Referenced by nd6_ra_input().

6.72.2.6 u_int32_t nd_ifinfo::maxmtu

Definition at line 79 of file nd6.h.

Referenced by nd6_ra_input(), and nd6_setmtu0().

6.72.2.7 u_int8_t nd_ifinfo::randomid[8]

Definition at line 90 of file nd6.h.

Referenced by in6_get_tmpifid(), and in6_tmpaddrtimer().

6.72.2.8 u_int8_t nd_ifinfo::randomseed0[8]

Definition at line 88 of file nd6.h.

Referenced by in6_get_tmpifid(), and in6_tmpaddrtimer().

6.72.2.9 u_int8_t nd_ifinfo::randomseed1[8]

Definition at line 89 of file nd6.h.

Referenced by in6_get_tmpifid(), and in6_tmpaddrtimer().

6.72.2.10 u_int32_t nd_ifinfo::reachable

Definition at line 81 of file nd6.h.

Referenced by nd6_ra_input(), and nd6_slowtimo().

6.72.2.11 int nd_ifinfo::recalectm

Definition at line 84 of file nd6.h.

Referenced by nd6_ra_input(), and nd6_slowtimo().

6.72.2.12 u_int32_t nd_ifinfo::retrans

Definition at line 82 of file nd6.h.

Referenced by nd6_llinfo_timer(), and nd6_ra_input().

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

- [/usr/src/sys/netinet6/nd6.h](#)

6.73 nd_opts Union Reference

```
#include <nd6.h>
```

Data Fields

- nd_opt_hdr * [nd_opt_array](#) [8]
- struct {
 - nd_opt_hdr * [zero](#)
 - nd_opt_hdr * [src_lladdr](#)
 - nd_opt_hdr * [tgt_lladdr](#)
 - nd_opt_prefix_info * [pi_beg](#)
 - nd_opt_rd_hdr * [rh](#)
 - nd_opt_mtu * [mtu](#)
 - nd_opt_hdr * [search](#)
 - nd_opt_hdr * [last](#)
 - int [done](#)
 - nd_opt_prefix_info * [pi_end](#)
- } [nd_opt_each](#)

6.73.1 Detailed Description

Definition at line 354 of file nd6.h.

6.73.2 Field Documentation

6.73.2.1 int [nd_opts::done](#)

Definition at line 365 of file nd6.h.

6.73.2.2 struct nd_opt_hdr* [nd_opts::last](#)

Definition at line 364 of file nd6.h.

6.73.2.3 struct nd_opt_mtu* [nd_opts::mtu](#)

Definition at line 362 of file nd6.h.

6.73.2.4 struct nd_opt_hdr* [nd_opts::nd_opt_array](#)[8]

Definition at line 355 of file nd6.h.

6.73.2.5 struct { ... } [nd_opts::nd_opt_each](#)

6.73.2.6 struct nd_opt_prefix_info* [nd_opts::pi_beg](#)

Definition at line 360 of file nd6.h.

6.73.2.7 struct nd_opt_prefix_info* nd_opts::pi_end

Definition at line 366 of file nd6.h.

6.73.2.8 struct nd_opt_rd_hdr* nd_opts::rh

Definition at line 361 of file nd6.h.

6.73.2.9 struct nd_opt_hdr* nd_opts::search

Definition at line 363 of file nd6.h.

6.73.2.10 struct nd_opt_hdr* nd_opts::src_lladdr

Definition at line 358 of file nd6.h.

6.73.2.11 struct nd_opt_hdr* nd_opts::tgt_lladdr

Definition at line 359 of file nd6.h.

6.73.2.12 struct nd_opt_hdr* nd_opts::zero

Definition at line 357 of file nd6.h.

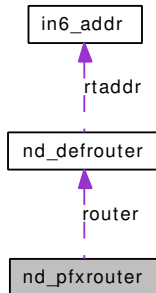
The documentation for this union was generated from the following file:

- [/usr/src/sys/netinet6/nd6.h](#)

6.74 nd_pfxrouter Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for nd_pfxrouter:



Public Member Functions

- [LIST_ENTRY \(nd_pfxrouter\)](#) pfr_entry

Data Fields

- [nd_defrouter](#) * router

6.74.1 Detailed Description

Definition at line 322 of file nd6.h.

6.74.2 Member Function Documentation

6.74.2.1 nd_pfxrouter::LIST_ENTRY (nd_pfxrouter)

6.74.3 Field Documentation

6.74.3.1 struct nd_defrouter* nd_pfxrouter::router

Definition at line 325 of file nd6.h.

Referenced by `find_pfxlist_reachable_router()`, `nd6_ioctl()`, `nd6_sysctl_prlist()`, and `pfxrtr_lookup()`.

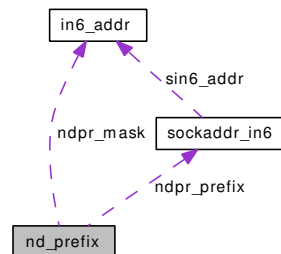
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/nd6.h`

6.75 nd_prefix Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for nd_prefix:



Public Member Functions

- [LIST_ENTRY](#) (`nd_prefix`) `ndpr_entry`
- [LIST_HEAD](#) (`pr_rtrhead`, `nd_pfxrouter`) `ndpr_advrtrs`

Data Fields

- `ifnet *` `ndpr_ifp`
- `sockaddr_in6` `ndpr_prefix`
- `in6_addr` `ndpr_mask`
- `u_int32_t` `ndpr_vltime`
- `u_int32_t` `ndpr_pltime`
- `time_t` `ndpr_expire`
- `time_t` `ndpr_preferred`
- `time_t` `ndpr_lastupdate`
- `prf_ra` `ndpr_flags`
- `u_int32_t` `ndpr_stateflags`
- `u_char` `ndpr_plen`
- `int` `ndpr_refcnt`

6.75.1 Detailed Description

Definition at line 268 of file `nd6.h`.

6.75.2 Member Function Documentation

6.75.2.1 `nd_prefix::LIST_ENTRY` ([nd_prefix](#))

6.75.2.2 `nd_prefix::LIST_HEAD` ([pr_rtrhead](#), [nd_pfxrouter](#))

6.75.3 Field Documentation

6.75.3.1 `time_t nd_prefix::ndpr_expire`

Definition at line 277 of file `nd6.h`.

Referenced by `in6_init_prefix_ltimes()`.

6.75.3.2 `struct prf_ra nd_prefix::ndpr_flags`

Definition at line 281 of file `nd6.h`.

6.75.3.3 `struct ifnet* nd_prefix::ndpr_ifp`

Definition at line 269 of file `nd6.h`.

Referenced by `nd6_ioctl()`, `nd6_is_new_addr_neighbor()`, `nd6_prefix_lookup()`, `nd6_prefix_offlink()`, `nd6_purge()`, `nd6_sysctl_prlist()`, and `prelist_update()`.

6.75.3.4 `time_t nd_prefix::ndpr_lastupdate`

Definition at line 279 of file `nd6.h`.

Referenced by `nd6_ioctl()`, `nd6_sysctl_prlist()`, and `prelist_update()`.

6.75.3.5 `struct in6_addr nd_prefix::ndpr_mask`

Definition at line 272 of file `nd6.h`.

Referenced by `nd6_is_new_addr_neighbor()`.

6.75.3.6 `u_char nd_prefix::ndpr_plen`

Definition at line 285 of file `nd6.h`.

Referenced by `nd6_ioctl()`, `nd6_prefix_lookup()`, `nd6_prefix_offlink()`, `nd6_prefix_onlink()`, `nd6_sysctl_prlist()`, `pxlist_onlink_check()`, and `prelist_update()`.

6.75.3.7 `u_int32_t nd_prefix::ndpr_pltime`

Definition at line 275 of file `nd6.h`.

Referenced by `in6_init_prefix_ltimes()`, `nd6_ioctl()`, `nd6_sysctl_prlist()`, and `prelist_update()`.

6.75.3.8 `time_t nd_prefix::ndpr_preferred`

Definition at line 278 of file nd6.h.

Referenced by `in6_init_prefix_ltimes()`.

6.75.3.9 `struct sockaddr_in6 nd_prefix::ndpr_prefix`

Definition at line 271 of file nd6.h.

Referenced by `nd6_ioctl()`, `nd6_is_new_addr_neighbor()`, `nd6_prefix_lookup()`, `nd6_prefix_offlink()`, `nd6_prefix_onlink()`, `nd6_sysctl_prlist()`, `pfxlist_onlink_check()`, and `prelist_update()`.

6.75.3.10 `int nd_prefix::ndpr_refcnt`

Definition at line 286 of file nd6.h.

Referenced by `in6_control()`, `in6_unlink_ifa()`, `nd6_purge()`, `nd6_sysctl_prlist()`, and `prelist_update()`.

6.75.3.11 `u_int32_t nd_prefix::ndpr_stateflags`

Definition at line 282 of file nd6.h.

Referenced by `nd6_is_new_addr_neighbor()`, `nd6_prefix_offlink()`, `nd6_prefix_onlink()`, `nd6_sysctl_prlist()`, `pfxlist_onlink_check()`, and `prelist_update()`.

6.75.3.12 `u_int32_t nd_prefix::ndpr_vltime`

Definition at line 274 of file nd6.h.

Referenced by `in6_init_prefix_ltimes()`, `nd6_ioctl()`, `nd6_sysctl_prlist()`, and `prelist_update()`.

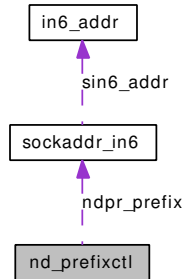
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/nd6.h`

6.76 nd_prefixctl Struct Reference

```
#include <nd6.h>
```

Collaboration diagram for nd_prefixctl:



Data Fields

- `ifnet *` `ndpr_ifp`
- `sockaddr_in6` `ndpr_prefix`
- `u_char` `ndpr_plen`
- `u_int32_t` `ndpr_vltime`
- `u_int32_t` `ndpr_pltime`
- `prf_ra` `ndpr_flags`

6.76.1 Detailed Description

Definition at line 254 of file `nd6.h`.

6.76.2 Field Documentation

6.76.2.1 struct `prf_ra` `nd_prefixctl::ndpr_flags`

Definition at line 264 of file `nd6.h`.

6.76.2.2 struct `ifnet*` `nd_prefixctl::ndpr_ifp`

Definition at line 255 of file `nd6.h`.

6.76.2.3 `u_char` `nd_prefixctl::ndpr_plen`

Definition at line 259 of file `nd6.h`.

6.76.2.4 `u_int32_t` `nd_prefixctl::ndpr_pltime`

Definition at line 262 of file `nd6.h`.

6.76.2.5 struct [sockaddr_in6](#) [nd_prefixctl::ndpr_prefix](#)

Definition at line 258 of file nd6.h.

6.76.2.6 u_int32_t [nd_prefixctl::ndpr_vltime](#)

Definition at line 261 of file nd6.h.

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

- [/usr/src/sys/netinet6/nd6.h](#)

6.77 newah Struct Reference

```
#include <ah.h>
```

Data Fields

- [u_int8_t ah_nxt](#)
- [u_int8_t ah_len](#)
- [u_int16_t ah_reserve](#)
- [u_int32_t ah_spi](#)
- [u_int32_t ah_seq](#)

6.77.1 Detailed Description

Definition at line 52 of file ah.h.

6.77.2 Field Documentation

6.77.2.1 u_int8_t newah::ah_len

Definition at line 54 of file ah.h.

6.77.2.2 u_int8_t newah::ah_nxt

Definition at line 53 of file ah.h.

6.77.2.3 u_int16_t newah::ah_reserve

Definition at line 55 of file ah.h.

6.77.2.4 u_int32_t newah::ah_seq

Definition at line 57 of file ah.h.

6.77.2.5 u_int32_t newah::ah_spi

Definition at line 56 of file ah.h.

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

- [/usr/src/sys/netinet6/ah.h](#)

6.78 newesp Struct Reference

```
#include <esp.h>
```

Data Fields

- [u_int32_t esp_spi](#)
- [u_int32_t esp_seq](#)

6.78.1 Detailed Description

Definition at line 55 of file esp.h.

6.78.2 Field Documentation

6.78.2.1 [u_int32_t newesp::esp_seq](#)

Definition at line 57 of file esp.h.

6.78.2.2 [u_int32_t newesp::esp_spi](#)

Definition at line 56 of file esp.h.

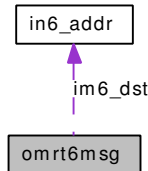
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/esp.h](#)

6.79 omrt6msg Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for omrt6msg:



Data Fields

- u_long [unused1](#)
- u_char [im6_msgtype](#)
- u_char [im6_mbz](#)
- u_char [im6_mif](#)
- u_char [unused2](#)
- [in6_addr](#) [im6_src](#) [im6_dst](#)

6.79.1 Detailed Description

Definition at line 148 of file ip6_mroute.h.

6.79.2 Field Documentation

6.79.2.1 struct [in6_addr](#) [im6_src](#) [omrt6msg::im6_dst](#)

Definition at line 159 of file ip6_mroute.h.

6.79.2.2 u_char [omrt6msg::im6_mbz](#)

Definition at line 156 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [ip6_mforward\(\)](#).

6.79.2.3 u_char [omrt6msg::im6_mif](#)

Definition at line 157 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [ip6_mforward\(\)](#).

6.79.2.4 u_char [omrt6msg::im6_msgtype](#)

Definition at line 150 of file ip6_mroute.h.

Referenced by [ip6_mdq\(\)](#), and [ip6_mforward\(\)](#).

6.79.2.5 u_long omrt6msg::unused1

Definition at line 149 of file ip6_mroute.h.

6.79.2.6 u_char omrt6msg::unused2

Definition at line 158 of file ip6_mroute.h.

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

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.80 pim Struct Reference

```
#include <pim6.h>
```

Data Fields

- u_char [pim_ver](#):4
- u_char [pim_type](#):4
- u_char [pim_rsv](#)
- u_short [pim_cksum](#)

6.80.1 Detailed Description

Definition at line 44 of file pim6.h.

6.80.2 Field Documentation

6.80.2.1 u_short [pim::pim_cksum](#)

Definition at line 57 of file pim6.h.

6.80.2.2 u_char [pim::pim_rsv](#)

Definition at line 56 of file pim6.h.

6.80.2.3 u_char [pim::pim_type](#)

Definition at line 53 of file pim6.h.

6.80.2.4 u_char [pim::pim_ver](#)

Definition at line 53 of file pim6.h.

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

- [/usr/src/sys/netinet6/pim6.h](#)

6.81 pim6stat Struct Reference

```
#include <pim6_var.h>
```

Data Fields

- `u_quad_t pim6s_rcv_total`
- `u_quad_t pim6s_rcv_tooshort`
- `u_quad_t pim6s_rcv_badsum`
- `u_quad_t pim6s_rcv_badversion`
- `u_quad_t pim6s_rcv_registers`
- `u_quad_t pim6s_rcv_badregisters`
- `u_quad_t pim6s_snd_registers`

6.81.1 Detailed Description

Definition at line 44 of file `pim6_var.h`.

6.81.2 Field Documentation

6.81.2.1 `u_quad_t pim6stat::pim6s_rcv_badregisters`

Definition at line 50 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.2 `u_quad_t pim6stat::pim6s_rcv_badsum`

Definition at line 47 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.3 `u_quad_t pim6stat::pim6s_rcv_badversion`

Definition at line 48 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.4 `u_quad_t pim6stat::pim6s_rcv_registers`

Definition at line 49 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.5 `u_quad_t pim6stat::pim6s_rcv_tooshort`

Definition at line 46 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.6 `u_quad_t pim6stat::pim6s_rev_total`

Definition at line 45 of file `pim6_var.h`.

Referenced by `pim6_input()`.

6.81.2.7 `u_quad_t pim6stat::pim6s_snd_registers`

Definition at line 51 of file `pim6_var.h`.

Referenced by `register_send()`.

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

- `/usr/src/sys/netinet6/pim6_var.h`

6.82 randomtab Struct Reference

Data Fields

- const int [ru_bits](#)
- const long [ru_out](#)
- const u_int32_t [ru_max](#)
- const u_int32_t [ru_gen](#)
- const u_int32_t [ru_n](#)
- const u_int32_t [ru_agen](#)
- const u_int32_t [ru_m](#)
- const u_int32_t [pfacts](#) [4]
- u_int32_t [ru_counter](#)
- u_int32_t [ru_msb](#)
- u_int32_t [ru_x](#)
- u_int32_t [ru_seed](#)
- u_int32_t [ru_seed2](#)
- u_int32_t [ru_a](#)
- u_int32_t [ru_b](#)
- u_int32_t [ru_g](#)
- long [ru_reseed](#)

6.82.1 Detailed Description

Definition at line 105 of file `ip6_id.c`.

6.82.2 Field Documentation

6.82.2.1 const u_int32_t [randomtab::pfacts](#)[4]

Definition at line 113 of file `ip6_id.c`.

Referenced by `initid()`.

6.82.2.2 u_int32_t [randomtab::ru_a](#)

Definition at line 120 of file `ip6_id.c`.

Referenced by `initid()`, and `randomid()`.

6.82.2.3 const u_int32_t [randomtab::ru_agen](#)

Definition at line 111 of file `ip6_id.c`.

Referenced by `initid()`.

6.82.2.4 u_int32_t [randomtab::ru_b](#)

Definition at line 120 of file `ip6_id.c`.

Referenced by `initid()`, and `randomid()`.

6.82.2.5 `const int randomtab::ru_bits`

Definition at line 106 of file ip6_id.c.

Referenced by `initid()`.

6.82.2.6 `u_int32_t randomtab::ru_counter`

Definition at line 115 of file ip6_id.c.

Referenced by `initid()`, and `randomid()`.

6.82.2.7 `u_int32_t randomtab::ru_g`

Definition at line 121 of file ip6_id.c.

Referenced by `initid()`, and `randomid()`.

6.82.2.8 `const u_int32_t randomtab::ru_gen`

Definition at line 109 of file ip6_id.c.

Referenced by `initid()`.

6.82.2.9 `const u_int32_t randomtab::ru_m`

Definition at line 112 of file ip6_id.c.

Referenced by `initid()`, and `randomid()`.

6.82.2.10 `const u_int32_t randomtab::ru_max`

Definition at line 108 of file ip6_id.c.

Referenced by `randomid()`.

6.82.2.11 `u_int32_t randomtab::ru_msb`

Definition at line 116 of file ip6_id.c.

Referenced by `initid()`, and `randomid()`.

6.82.2.12 `const u_int32_t randomtab::ru_n`

Definition at line 110 of file ip6_id.c.

Referenced by `initid()`, and `randomid()`.

6.82.2.13 `const long randomtab::ru_out`

Definition at line 107 of file ip6_id.c.

Referenced by `initid()`.

6.82.2.14 long [randomtab::ru_reseed](#)

Definition at line 122 of file ip6_id.c.

Referenced by [initid\(\)](#), and [randomid\(\)](#).

6.82.2.15 u_int32_t [randomtab::ru_seed](#)

Definition at line 119 of file ip6_id.c.

Referenced by [initid\(\)](#), and [randomid\(\)](#).

6.82.2.16 u_int32_t [randomtab::ru_seed2](#)

Definition at line 119 of file ip6_id.c.

Referenced by [initid\(\)](#), and [randomid\(\)](#).

6.82.2.17 u_int32_t [randomtab::ru_x](#)

Definition at line 118 of file ip6_id.c.

Referenced by [initid\(\)](#), and [randomid\(\)](#).

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

- [/usr/src/sys/netinet6/ip6_id.c](#)

6.83 rip6stat Struct Reference

```
#include <raw_ip6.h>
```

Data Fields

- `u_quad_t rip6s_ipackets`
- `u_quad_t rip6s_isum`
- `u_quad_t rip6s_badsum`
- `u_quad_t rip6s_nosock`
- `u_quad_t rip6s_nosockmcast`
- `u_quad_t rip6s_fullsock`
- `u_quad_t rip6s_opackets`

6.83.1 Detailed Description

Definition at line 39 of file `raw_ip6.h`.

6.83.2 Field Documentation

6.83.2.1 `u_quad_t rip6stat::rip6s_badsum`

Definition at line 42 of file `raw_ip6.h`.

Referenced by `rip6_input()`.

6.83.2.2 `u_quad_t rip6stat::rip6s_fullsock`

Definition at line 45 of file `raw_ip6.h`.

Referenced by `rip6_input()`.

6.83.2.3 `u_quad_t rip6stat::rip6s_ipackets`

Definition at line 40 of file `raw_ip6.h`.

Referenced by `rip6_input()`.

6.83.2.4 `u_quad_t rip6stat::rip6s_isum`

Definition at line 41 of file `raw_ip6.h`.

Referenced by `rip6_input()`.

6.83.2.5 `u_quad_t rip6stat::rip6s_nosock`

Definition at line 43 of file `raw_ip6.h`.

Referenced by `rip6_input()`.

6.83.2.6 u_quad_t rip6stat::rip6s_nosockmcast

Definition at line 44 of file raw_ip6.h.

Referenced by rip6_input().

6.83.2.7 u_quad_t rip6stat::rip6s_opackets

Definition at line 47 of file raw_ip6.h.

Referenced by rip6_output().

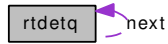
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/raw_ip6.h](#)

6.84 rtdetq Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for rtdetq:



Data Fields

- mbuf * [m](#)
- ifnet * [ifp](#)
- rtdetq * [next](#)

6.84.1 Detailed Description

Definition at line 252 of file ip6_mroute.h.

6.84.2 Field Documentation

6.84.2.1 struct ifnet* [rtdetq::ifp](#)

Definition at line 254 of file ip6_mroute.h.

Referenced by [add_m6fc\(\)](#).

6.84.2.2 struct mbuf* [rtdetq::m](#)

Definition at line 253 of file ip6_mroute.h.

Referenced by [add_m6fc\(\)](#), [expire_upcalls\(\)](#), [ip6_mforward\(\)](#), and [ip6_mrouter_done\(\)](#).

6.84.2.3 struct rtdetq* [rtdetq::next](#)

Definition at line 258 of file ip6_mroute.h.

Referenced by [add_m6fc\(\)](#), [expire_upcalls\(\)](#), [ip6_mforward\(\)](#), and [ip6_mrouter_done\(\)](#).

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

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.85 rtqk_arg Struct Reference

Data Fields

- radix_node_head * [mh](#)
- int [mode](#)
- int [updating](#)
- int [draining](#)
- int [killed](#)
- int [found](#)
- time_t [nextstop](#)

6.85.1 Detailed Description

Definition at line 268 of file in6_rmx.c.

6.85.2 Field Documentation

6.85.2.1 int [rtqk_arg::draining](#)

Definition at line 272 of file in6_rmx.c.

Referenced by [in6_rtqkill\(\)](#), and [in6_rtqtimeo\(\)](#).

6.85.2.2 int [rtqk_arg::found](#)

Definition at line 274 of file in6_rmx.c.

Referenced by [in6_rtqkill\(\)](#), and [in6_rtqtimeo\(\)](#).

6.85.2.3 int [rtqk_arg::killed](#)

Definition at line 273 of file in6_rmx.c.

Referenced by [in6_rtqkill\(\)](#), and [in6_rtqtimeo\(\)](#).

6.85.2.4 int [rtqk_arg::mode](#)

Definition at line 270 of file in6_rmx.c.

6.85.2.5 time_t [rtqk_arg::nextstop](#)

Definition at line 275 of file in6_rmx.c.

Referenced by [in6_rtqkill\(\)](#), and [in6_rtqtimeo\(\)](#).

6.85.2.6 struct radix_node_head* [rtqk_arg::rnh](#)

Definition at line 269 of file in6_rmx.c.

Referenced by [in6_rtqtimeo\(\)](#).

6.85.2.7 int [rtqk_arg::updating](#)

Definition at line 271 of file [in6_rmx.c](#).

Referenced by [in6_rtqkill\(\)](#), and [in6_rtqtimeo\(\)](#).

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

- [/usr/src/sys/netinet6/in6_rmx.c](#)

6.86 scope6_id Struct Reference

```
#include <scope6_var.h>
```

Data Fields

- `u_int32_t s6id_list` [16]

6.86.1 Detailed Description

Definition at line 37 of file `scope6_var.h`.

6.86.2 Field Documentation

6.86.2.1 `u_int32_t scope6_id::s6id_list`[16]

Definition at line 42 of file `scope6_var.h`.

Referenced by `in6_setscope()`, `scope6_addr2default()`, `scope6_set()`, and `scope6_setdefault()`.

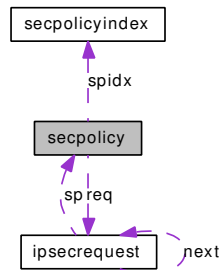
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/scope6_var.h`

6.87 secpolicy Struct Reference

```
#include <ipsec.h>
```

Collaboration diagram for secpolicy:



Public Member Functions

- [TAILQ_ENTRY \(secpolicy\)](#) tailq
- [LIST_ENTRY \(secpolicy\)](#) chain

Data Fields

- `u_int8_t dir`
- `int readonly`
- `int persist`
- `int refcnt`
- `secpolicyindex * spidx`
- `u_int32_t id`
- `socket * so`
- `u_int state`
- `int policy`
- `ipsecrequest * req`
- `long created`
- `long lastused`
- `long lifetime`
- `long validtime`

6.87.1 Detailed Description

Definition at line 71 of file `ipsec.h`.

6.87.2 Member Function Documentation

6.87.2.1 secpolicy::LIST_ENTRY (secpolicy)

6.87.2.2 secpolicy::TAILQ_ENTRY (secpolicy)

6.87.3 Field Documentation

6.87.3.1 long secpolicy::created

Definition at line 103 of file ipsec.h.

6.87.3.2 u_int8_t secpolicy::dir

Definition at line 75 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), and ipsec_init_pcbpolicy().

6.87.3.3 u_int32_t secpolicy::id

Definition at line 80 of file ipsec.h.

6.87.3.4 long secpolicy::lastused

Definition at line 104 of file ipsec.h.

Referenced by ipsec_checkpcbcache().

6.87.3.5 long secpolicy::lifetime

Definition at line 105 of file ipsec.h.

6.87.3.6 int secpolicy::persist

Definition at line 77 of file ipsec.h.

Referenced by ipsec_init_pcbpolicy().

6.87.3.7 int secpolicy::policy

Definition at line 91 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec4_getpolicybypcb(), ipsec_deepcopy_policy(), and ipsec_init_pcbpolicy().

6.87.3.8 int secpolicy::readonly

Definition at line 76 of file ipsec.h.

Referenced by ipsec_init_pcbpolicy().

6.87.3.9 int [secpolicy::refcnt](#)

Definition at line 78 of file ipsec.h.

Referenced by ipsec4_getpolicybyaddr(), ipsec4_getpolicybypcb(), ipsec_checkpcbcache(), ipsec_fillpcbcache(), and ipsec_init_pcbpolicy().

6.87.3.10 struct [ipsecrequest*](#) [secpolicy::req](#)

Definition at line 92 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), and ipsec_deepcopy_policy().

6.87.3.11 struct [socket*](#) [secpolicy::so](#)

Definition at line 86 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), and ipsec_init_pcbpolicy().

6.87.3.12 struct [secpolicyindex*](#) [secpolicy::spidx](#)

Definition at line 79 of file ipsec.h.

Referenced by ipsec_checkpcbcache(), and ipsec_deepcopy_policy().

6.87.3.13 u_int [secpolicy::state](#)

Definition at line 87 of file ipsec.h.

Referenced by ipsec_checkpcbcache(), ipsec_deepcopy_policy(), ipsec_init_pcbpolicy(), and ipsec_set_policy().

6.87.3.14 long [secpolicy::validtime](#)

Definition at line 106 of file ipsec.h.

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.88 secpolicyindex Struct Reference

```
#include <ipsec.h>
```

Data Fields

- sockaddr_storage [src](#)
- sockaddr_storage [dst](#)
- u_int8_t [prefs](#)
- u_int8_t [prefd](#)
- u_int16_t [ul_proto](#)

6.88.1 Detailed Description

Definition at line 56 of file ipsec.h.

6.88.2 Field Documentation

6.88.2.1 struct sockaddr_storage [secpolicyindex::dst](#)

Definition at line 58 of file ipsec.h.

Referenced by ipsec4_get_ulp(), and ipsec4_setspidx_ipaddr().

6.88.2.2 u_int8_t [secpolicyindex::prefd](#)

Definition at line 60 of file ipsec.h.

Referenced by ipsec4_setspidx_ipaddr().

6.88.2.3 u_int8_t [secpolicyindex::prefs](#)

Definition at line 59 of file ipsec.h.

Referenced by ipsec4_setspidx_ipaddr().

6.88.2.4 struct sockaddr_storage [secpolicyindex::src](#)

Definition at line 57 of file ipsec.h.

Referenced by ipsec4_get_ulp(), and ipsec4_setspidx_ipaddr().

6.88.2.5 u_int16_t [secpolicyindex::ul_proto](#)

Definition at line 61 of file ipsec.h.

Referenced by ipsec4_get_ulp().

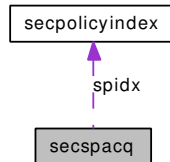
The documentation for this struct was generated from the following file:

- /usr/src/sys/netinet6/ipsec.h

6.89 secspacq Struct Reference

```
#include <ipsec.h>
```

Collaboration diagram for secspacq:



Public Member Functions

- [LIST_ENTRY \(secspacq\)](#) chain

Data Fields

- [secpolicyindex spidx](#)
- long [created](#)
- int [count](#)

6.89.1 Detailed Description

Definition at line 141 of file ipsec.h.

6.89.2 Member Function Documentation

6.89.2.1 [secspacq::LIST_ENTRY \(secspacq\)](#)

6.89.3 Field Documentation

6.89.3.1 int [secspacq::count](#)

Definition at line 147 of file ipsec.h.

6.89.3.2 long [secspacq::created](#)

Definition at line 146 of file ipsec.h.

6.89.3.3 struct [secpolicyindex secspacq::spidx](#)

Definition at line 144 of file ipsec.h.

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

- [/usr/src/sys/netinet6/ipsec.h](#)

6.90 sioc_mif_req6 Struct Reference

```
#include <ip6_mroute.h>
```

Data Fields

- [mifi_t mifi](#)
- [u_quad_t icount](#)
- [u_quad_t ocount](#)
- [u_quad_t ibytes](#)
- [u_quad_t obytes](#)

6.90.1 Detailed Description

Definition at line 197 of file ip6_mroute.h.

6.90.2 Field Documentation

6.90.2.1 [u_quad_t sioc_mif_req6::ibytes](#)

Definition at line 201 of file ip6_mroute.h.

6.90.2.2 [u_quad_t sioc_mif_req6::icount](#)

Definition at line 199 of file ip6_mroute.h.

6.90.2.3 [mifi_t sioc_mif_req6::mifi](#)

Definition at line 198 of file ip6_mroute.h.

6.90.2.4 [u_quad_t sioc_mif_req6::obytes](#)

Definition at line 202 of file ip6_mroute.h.

6.90.2.5 [u_quad_t sioc_mif_req6::ocount](#)

Definition at line 200 of file ip6_mroute.h.

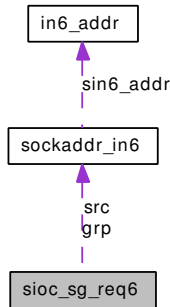
The documentation for this struct was generated from the following file:

- [/usr/src/sys/netinet6/ip6_mroute.h](#)

6.91 sioc_sg_req6 Struct Reference

```
#include <ip6_mroute.h>
```

Collaboration diagram for sioc_sg_req6:



Data Fields

- [sockaddr_in6 src](#)
- [sockaddr_in6 grp](#)
- [u_quad_t pktcnt](#)
- [u_quad_t bytecnt](#)
- [u_quad_t wrong_if](#)

6.91.1 Detailed Description

Definition at line 186 of file ip6_mroute.h.

6.91.2 Field Documentation

6.91.2.1 u_quad_t sioc_sg_req6::bytecnt

Definition at line 190 of file ip6_mroute.h.

6.91.2.2 struct sockaddr_in6 sioc_sg_req6::grp

Definition at line 188 of file ip6_mroute.h.

6.91.2.3 u_quad_t sioc_sg_req6::pktcnt

Definition at line 189 of file ip6_mroute.h.

6.91.2.4 struct sockaddr_in6 sioc_sg_req6::src

Definition at line 187 of file ip6_mroute.h.

6.91.2.5 `u_quad_t` `sioc_sg_req6::wrong_if`

Definition at line 191 of file `ip6_mroute.h`.

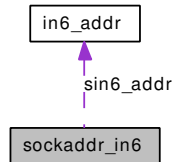
The documentation for this struct was generated from the following file:

- `/usr/src/sys/netinet6/ip6_mroute.h`

6.92 sockaddr_in6 Struct Reference

```
#include <in6.h>
```

Collaboration diagram for `sockaddr_in6`:



Data Fields

- `uint8_t` [sin6_len](#)
- `sa_family_t` [sin6_family](#)
- `in_port_t` [sin6_port](#)
- `uint32_t` [sin6_flowinfo](#)
- `in6_addr` [sin6_addr](#)
- `uint32_t` [sin6_scope_id](#)

6.92.1 Detailed Description

Definition at line 123 of file `in6.h`.

6.92.2 Field Documentation

6.92.2.1 struct `in6_addr` `sockaddr_in6::sin6_addr`

Definition at line 128 of file `in6.h`.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, `gif_validate6()`, `icmp6_redirect_output()`, `icmp6_reflect()`, `in6_addroute()`, `in6_control()`, `in6_gif_output()`, `in6_ifadd()`, `in6_ifinit()`, `in6_ifremloop()`, `in6_is_addr_deprecated()`, `in6_lifaddr_ioctl()`, `in6_localaddr()`, `in6_pcbconnect()`, `in6_pcbaddr()`, `in6_pcbnotify()`, `in6_purgeaddr()`, `in6_selectsrc()`, `in6_sin6_2_sin()`, `in6_src_ioctl()`, `in6_update_ifa()`, `ip6_forward()`, `ip6_getpmtu()`, `ip6_input()`, `ip6_mdq()`, `ip6_mforward()`, `ip6_setpktopt()`, `mld6_sendpkt()`, `nd6_dad_duplicated()`, `nd6_dad_ns_input()`, `nd6_dad_ns_output()`, `nd6_dad_start()`, `nd6_dad_timer()`, `nd6_init()`, `nd6_ioctl()`, `nd6_is_new_addr_neighbor()`, `nd6_llinfo_timer()`, `nd6_na_input()`, `nd6_na_output()`, `nd6_output()`, `nd6_prefix_lookup()`, `nd6_prefix_offlink()`, `nd6_prefix_onlink()`, `ni6_addrs()`, `ni6_store_addrs()`, `pfxlist_onlink_check()`, `phyint_send()`, `prelist_update()`, `register_send()`, `rip6_output()`, `sctp6_bind()`, `sctp6_connect()`, `sctp6_getaddr()`, `sctp6_peeraddr()`, `sctp6_send()`, `selectroute()`, `udp6_bind()`, `udp6_connect()`, `udp6_output()`, and `udp6_send()`.

6.92.2.2 `sa_family_t` `sockaddr_in6::sin6_family`

Definition at line 125 of file `in6.h`.

Referenced by `in6_control()`, `in6_gif_output()`, `in6_ifinit()`, `in6_pcbaddr()`, `in6_update_ifa()`, `ip6_input()`, `ip6_mloopback()`, `ip6_setmoptions()`, `nd6_init()`, `nd6_na_output()`, `phyint_send()`, `pim6_input()`, `rip6_send()`, `sctp6_getaddr()`, `sctp6_peeraddr()`, and `selectroute()`.

6.92.2.3 uint32_t sockaddr_in6::sin6_flowinfo

Definition at line 127 of file in6.h.

Referenced by in6_pcbnotify().

6.92.2.4 uint8_t sockaddr_in6::sin6_len

Definition at line 124 of file in6.h.

Referenced by in6_control(), in6_lifaddr_ioctl(), in6_purgeaddr(), in6_update_ifa(), ip6_input(), ip6_setpktopt(), nd6_init(), nd6_na_output(), and phyint_send().

6.92.2.5 in_port_t sockaddr_in6::sin6_port

Definition at line 126 of file in6.h.

Referenced by in6_pcbbind(), in6_pcbconnect(), in6_pcblladdr(), in6_sin6_2_sin(), sctp6_peeraddr(), udp6_input(), and udp6_output().

6.92.2.6 uint32_t sockaddr_in6::sin6_scope_id

Definition at line 129 of file in6.h.

Referenced by in6_control(), in6_pcblladdr(), in6_update_ifa(), nd6_is_new_addr_neighbor(), rip6_output(), selectroute(), and udp6_output().

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

- [/usr/src/sys/netinet6/in6.h](#)

6.93 walkarg Struct Reference

Data Fields

- sysctl_req * [w_req](#)

6.93.1 Detailed Description

Definition at line 884 of file [in6_src.c](#).

6.93.2 Field Documentation

6.93.2.1 struct sysctl_req* [walkarg::w_req](#)

Definition at line 885 of file [in6_src.c](#).

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

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

- [/usr/src/sys/netinet6/in6_src.c](#)

Chapter 7

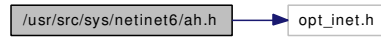
FreeBSD kernel IPv6 code File Documentation

7.1 notreviewed.dox File Reference

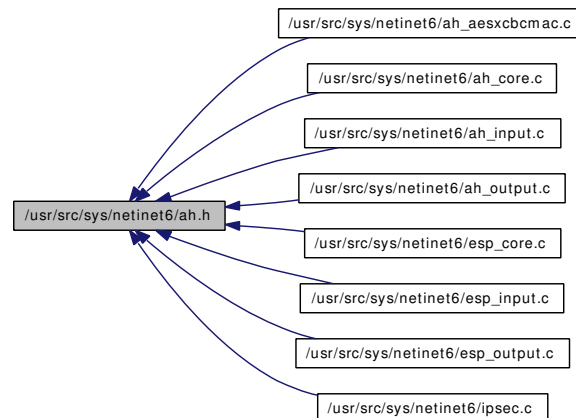
7.2 /usr/src/sys/netinet6/ah.h File Reference

```
#include "opt_inet.h"
```

Include dependency graph for ah.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ah](#)
- struct [newah](#)
- struct [ah_algorithm_state](#)
- struct [ah_algorithm](#)

Defines

- #define [AH_MAXSUMSIZE](#) (512 / 8)

Functions

- [ah_algorithm](#) *ah_algorithm_lookup __P ((int))
- int ah_hdrlen __P ((struct secasvar *))
- [size_t](#) ah_hdrsiz __P ((struct ipsecrequest *))
- void ah4_input __P ((struct mbuf *, int))
- int ah4_output __P ((struct mbuf *, struct ipsecrequest *))
- int ah4_calcksum __P ((struct mbuf *, u_int8_t *, [size_t](#), const struct [ah_algorithm](#) *, struct secasvar *))

7.2.1 Define Documentation

7.2.1.1 #define AH_MAXSUMSIZE (512 / 8)

Definition at line 80 of file ah.h.

Referenced by ah_hdrsiz(), and esp_auth().

7.2.2 Function Documentation

7.2.2.1 int ah4_calchecksum __P ((struct mbuf *, u_int8_t *, size_t, const struct ah_algorithm *, struct secasvar *))

7.2.2.2 int ah4_output __P ((struct mbuf *, struct ipsecrequest *))

7.2.2.3 void ip6_forward __P ((struct mbuf *, int))

7.2.2.4 size_t ah_hdrsiz __P ((struct ipsecrequest *))

7.2.2.5 static int esp_cbc_mature __P ((struct secasvar *))

7.2.2.6 struct ah_algorithm* ah_algorithm_lookup __P ((int))

Referenced by icmp6_notify_error(), mld_starttimer(), nd6_dad_starttimer(), rip6_ctlinput(), and udp6_ctlinput().

7.3 /usr/src/sys/netinet6/ah6.h File Reference

Functions

- int ah6_input __P ((struct mbuf **, int *, int))
- int ah6_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))
- int ah6_calchecksum __P ((struct mbuf *, u_int8_t *, size_t, const struct ah_algorithm *, struct secasvar *))
- void ah6_ctlinput __P ((int, struct sockaddr *, void *))

7.3.1 Function Documentation

7.3.1.1 void ah6_ctlinput __P ((int, struct sockaddr *, void *))

7.3.1.2 int ah6_calchecksum __P ((struct mbuf *, u_int8_t *, size_t, const struct ah_algorithm *, struct secasvar *))

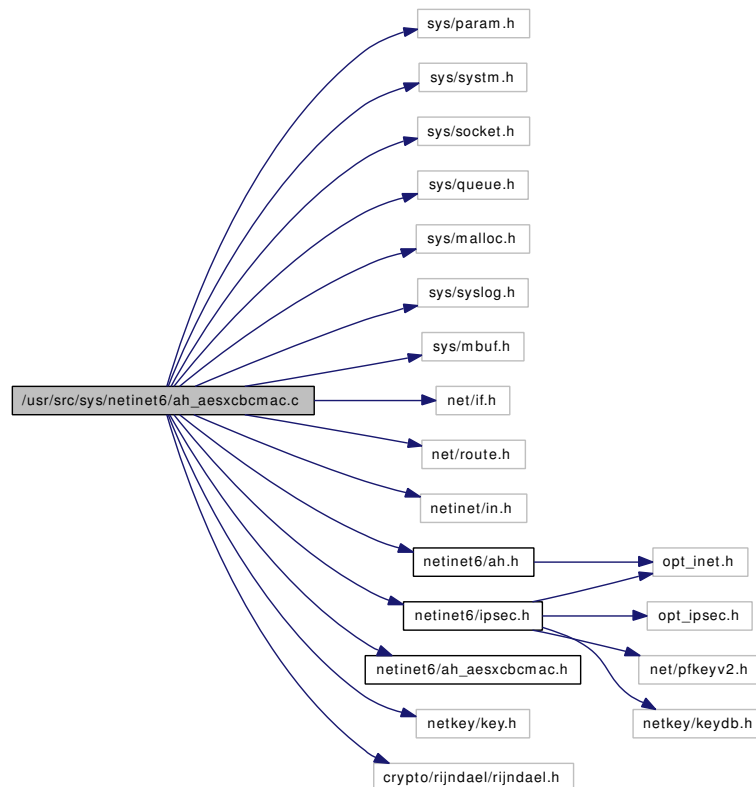
7.3.1.3 int ah6_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))

7.3.1.4 int none_input __P ((struct mbuf **, int *, int))

7.4 /usr/src/sys/netinet6/ah_aesxcbcmac.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/socket.h>
#include <sys/queue.h>
#include <sys/malloc.h>
#include <sys/syslog.h>
#include <sys/mbuf.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet6/ipsec.h>
#include <netinet6/ah.h>
#include <netinet6/ah_aesxcbcmac.h>
#include <netkey/key.h>
#include <crypto/rijndael/rijndael.h>
```

Include dependency graph for ah_aesxcbcmac.c:



Data Structures

- struct [aesxcbc_ctx](#)

Defines

- #define [AES_BLOCKSIZE](#) 16

Functions

- int [ah_aes_xcbc_mac_init](#) (struct [ah_algorithm_state](#) *state, struct [secasvar](#) *sav)
- void [ah_aes_xcbc_mac_loop](#) (struct [ah_algorithm_state](#) *state, u_int8_t *addr, [size_t](#) len)
- void [ah_aes_xcbc_mac_result](#) (struct [ah_algorithm_state](#) *state, u_int8_t *addr, [size_t](#) l)

7.4.1 Define Documentation

7.4.1.1 #define AES_BLOCKSIZE 16

Definition at line 55 of file [ah_aesxcbcmac.c](#).

Referenced by [ah_aes_xcbc_mac_init\(\)](#), [ah_aes_xcbc_mac_loop\(\)](#), [ah_aes_xcbc_mac_result\(\)](#), [esp_aesctr_decrypt\(\)](#), and [esp_aesctr_encrypt\(\)](#).

7.4.2 Function Documentation

7.4.2.1 int ah_aes_xcbc_mac_init (struct [ah_algorithm_state](#) *state, struct [secasvar](#) *sav)

Definition at line 70 of file [ah_aesxcbcmac.c](#).

References [AES_BLOCKSIZE](#).

7.4.2.2 void ah_aes_xcbc_mac_loop (struct [ah_algorithm_state](#) *state, u_int8_t *addr, [size_t](#) len)

Definition at line 109 of file [ah_aesxcbcmac.c](#).

References [AES_BLOCKSIZE](#), [aesxcbc_ctx::buf](#), [aesxcbc_ctx::buflen](#), [aesxcbc_ctx::e](#), [aesxcbc_ctx::r_k1s](#), and [aesxcbc_ctx::r_nr](#).

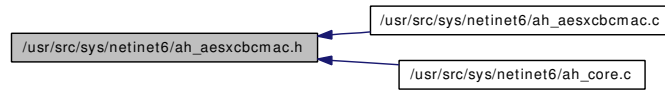
7.4.2.3 void ah_aes_xcbc_mac_result (struct [ah_algorithm_state](#) *state, u_int8_t *addr, [size_t](#) l)

Definition at line 160 of file [ah_aesxcbcmac.c](#).

References [AES_BLOCKSIZE](#), [aesxcbc_ctx::buf](#), [aesxcbc_ctx::buflen](#), [aesxcbc_ctx::e](#), [aesxcbc_ctx::k2](#), [aesxcbc_ctx::k3](#), [aesxcbc_ctx::r_k1s](#), and [aesxcbc_ctx::r_nr](#).

7.5 /usr/src/sys/netinet6/ah_aesxcbcmac.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int ah_aes_xcbc_mac_init __P ((struct ah_algorithm_state *, struct secasvar *))`
- `void ah_aes_xcbc_mac_loop __P ((struct ah_algorithm_state *, u_int8_t *, size_t))`
- `void ah_aes_xcbc_mac_result __P ((struct ah_algorithm_state *, u_int8_t *, size_t))`

7.5.1 Function Documentation

7.5.1.1 `static void ah_hmac_ripemd160_result __P ((struct ah_algorithm_state *, u_int8_t *, size_t))`

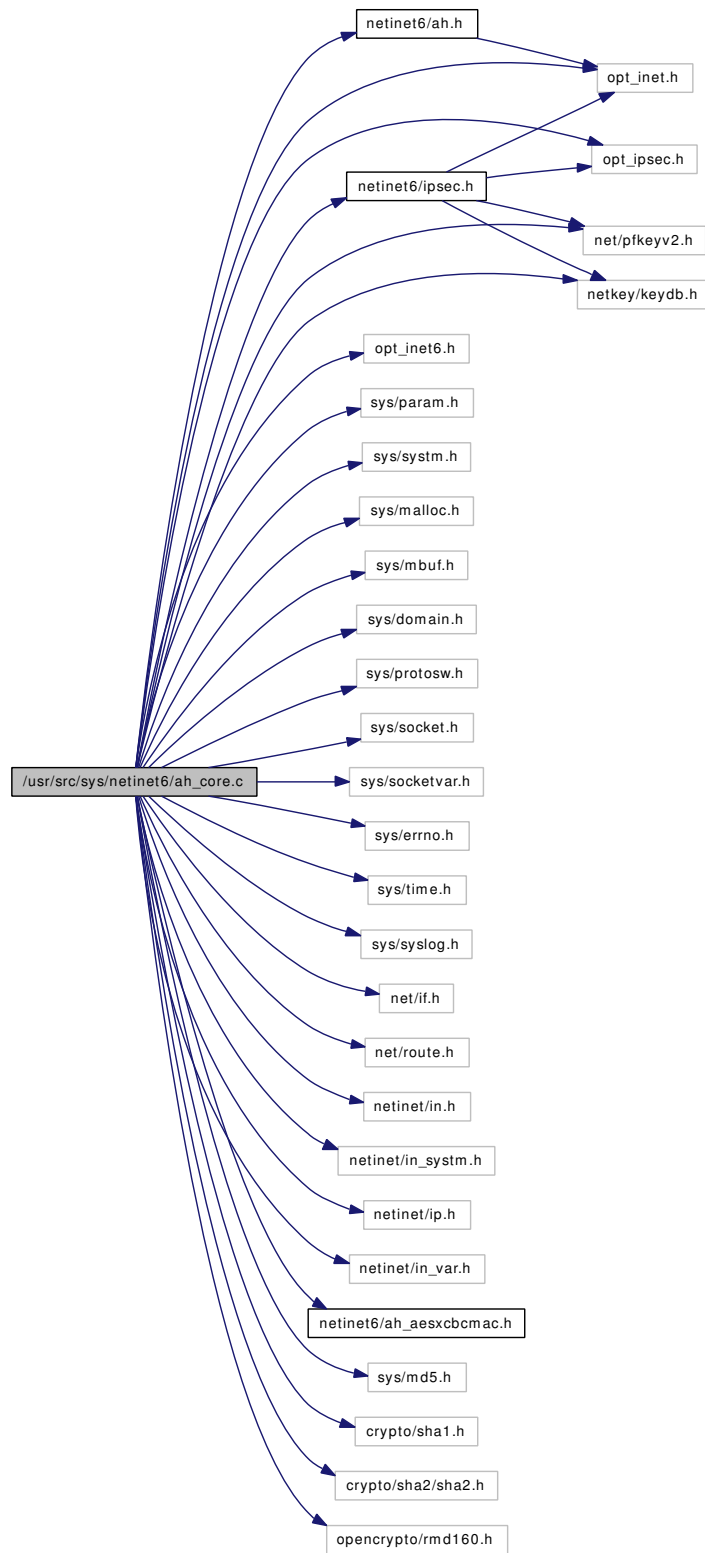
7.5.1.2 `static void ah_hmac_ripemd160_loop __P ((struct ah_algorithm_state *, u_int8_t *, size_t))`

7.5.1.3 `static int ah_hmac_ripemd160_init __P ((struct ah_algorithm_state *, struct secasvar *))`

7.6 /usr/src/sys/netinet6/ah_core.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/in_var.h>
#include <netinet6/ipsec.h>
#include <netinet6/ah.h>
#include <netinet6/ah_aesxcbcmac.h>
#include <net/pfkeyv2.h>
#include <netkey/keydb.h>
#include <sys/md5.h>
#include <crypto/sha1.h>
#include <crypto/sha2/sha2.h>
#include <opencrypto/rmd160.h>
```

Include dependency graph for ah_core.c:



Defines

- #define MD5_RESULTLEN 16
- #define RIPEMD160_RESULTLEN 20

Functions

- static int ah_sumsiz_1216 __P((struct secasvar *))
- static int ah_none_init __P((struct ah_algorithm_state *, struct secasvar *))
- static void ah_none_loop __P((struct ah_algorithm_state *, u_int8_t *, size_t))
- static void ah_none_result __P((struct ah_algorithm_state *, u_int8_t *, size_t))
- static int ah_keyed_md5_init __P((struct ah_algorithm_state *, struct secasvar *))
- static void ah_keyed_md5_loop __P((struct ah_algorithm_state *, u_int8_t *, size_t))
- static void ah_update_mbuf __P((struct mbuf *, int, int, const struct ah_algorithm *, struct ah_algorithm_state *))
- ah_algorithm * ah_algorithm_lookup (int idx)
- static int ah_sumsiz_1216 (struct secasvar *sav)
- static int ah_sumsiz_zero (struct secasvar *sav)
- static int ah_common_mature (struct secasvar *sav)
- static int ah_none_mature (struct secasvar *sav)
- static int ah_none_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_none_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_none_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_keyed_md5_mature (struct secasvar *sav)
- static int ah_keyed_md5_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_keyed_md5_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_keyed_md5_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_keyed_sha1_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_keyed_sha1_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_keyed_sha1_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_md5_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_md5_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_md5_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_sha1_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_sha1_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_sha1_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_sha2_256_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_sha2_256_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_sha2_256_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_sha2_384_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_sha2_384_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_sha2_384_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_sha2_512_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_sha2_512_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_sha2_512_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static int ah_hmac_ripemd160_init (struct ah_algorithm_state *state, struct secasvar *sav)
- static void ah_hmac_ripemd160_loop (struct ah_algorithm_state *state, u_int8_t *addr, size_t len)
- static void ah_hmac_ripemd160_result (struct ah_algorithm_state *state, u_int8_t *addr, size_t l)
- static void ah_update_mbuf (struct mbuf *m, int off, int len, const struct ah_algorithm *algo, struct ah_algorithm_state *algorithms)

Variables

- static struct [ah_algorithm](#) [ah_algorithms](#) []

7.6.1 Define Documentation

7.6.1.1 #define MD5_RESULTLEN 16

Definition at line 88 of file [ah_core.c](#).

Referenced by [ah_hmac_md5_init\(\)](#), [ah_hmac_md5_result\(\)](#), and [ah_keyed_md5_result\(\)](#).

7.6.1.2 #define RIPEMD160_RESULTLEN 20

Definition at line 92 of file [ah_core.c](#).

Referenced by [ah_hmac_ripemd160_init\(\)](#), and [ah_hmac_ripemd160_result\(\)](#).

7.6.2 Function Documentation

7.6.2.1 `static void ah_update_mbuf __P ((struct mbuf *, int, int, const struct ah_algorithm *, struct ah_algorithm_state *))` [static]

7.6.2.2 `static void ah_keyed_md5_loop __P ((struct ah_algorithm_state *, u_int8_t *, size_t))` [static]

7.6.2.3 `static int ah_keyed_md5_init __P ((struct ah_algorithm_state *, struct secasvar *))` [static]

7.6.2.4 `static void ah_none_result __P ((struct ah_algorithm_state *, u_int8_t *, size_t))` [static]

7.6.2.5 `static void ah_none_loop __P ((struct ah_algorithm_state *, u_int8_t *, size_t))` [static]

7.6.2.6 `static int ah_none_init __P ((struct ah_algorithm_state *, struct secasvar *))` [static]

7.6.2.7 `static int ah_sumsiz_1216 __P ((struct secasvar *))` [static]

7.6.2.8 `struct ah_algorithm* ah_algorithm_lookup (int idx)`

Definition at line 198 of file [ah_core.c](#).

References [ah_algorithms](#).

Referenced by [ah_common_mature\(\)](#), [ah_hdrlen\(\)](#), [ah_hdrsiz\(\)](#), [esp_auth\(\)](#), and [esp_hdrsiz\(\)](#).

7.6.2.9 `static int ah_common_mature (struct secasvar * sav)` [static]

Definition at line 253 of file [ah_core.c](#).

References [ah_algorithm_lookup\(\)](#), [ipseclog](#), [ah_algorithm::keymax](#), [ah_algorithm::keymin](#), and [ah_algorithm::name](#).

Here is the call graph for this function:



7.6.2.10 `static int ah_hmac_md5_init (struct ah_algorithm_state * state, struct secasvar * sav)`
[static]

Definition at line 513 of file ah_core.c.

References MD5_RESULTLEN.

7.6.2.11 `static void ah_hmac_md5_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len)` [static]

Definition at line 566 of file ah_core.c.

7.6.2.12 `static void ah_hmac_md5_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l)` [static]

Definition at line 580 of file ah_core.c.

References MD5_RESULTLEN.

7.6.2.13 `static int ah_hmac_ripemd160_init (struct ah_algorithm_state * state, struct secasvar * sav)` [static]

Definition at line 1020 of file ah_core.c.

References RIPEMD160_RESULTLEN.

7.6.2.14 `static void ah_hmac_ripemd160_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len)` [static]

Definition at line 1078 of file ah_core.c.

7.6.2.15 `static void ah_hmac_ripemd160_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l)` [static]

Definition at line 1093 of file ah_core.c.

References RIPEMD160_RESULTLEN.

7.6.2.16 `static int ah_hmac_sha1_init (struct ah_algorithm_state * state, struct secasvar * sav)`
[static]

Definition at line 610 of file ah_core.c.

7.6.2.17 `static void ah_hmac_sha1_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 664 of file ah_core.c.

7.6.2.18 `static void ah_hmac_sha1_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 679 of file ah_core.c.

7.6.2.19 `static int ah_hmac_sha2_256_init (struct ah_algorithm_state * state, struct secasvar * sav) [static]`

Definition at line 709 of file ah_core.c.

7.6.2.20 `static void ah_hmac_sha2_256_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 766 of file ah_core.c.

7.6.2.21 `static void ah_hmac_sha2_256_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 781 of file ah_core.c.

7.6.2.22 `static int ah_hmac_sha2_384_init (struct ah_algorithm_state * state, struct secasvar * sav) [static]`

Definition at line 812 of file ah_core.c.

7.6.2.23 `static void ah_hmac_sha2_384_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 870 of file ah_core.c.

7.6.2.24 `static void ah_hmac_sha2_384_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 885 of file ah_core.c.

7.6.2.25 `static int ah_hmac_sha2_512_init (struct ah_algorithm_state * state, struct secasvar * sav) [static]`

Definition at line 916 of file ah_core.c.

7.6.2.26 `static void ah_hmac_sha2_512_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 974 of file `ah_core.c`.

7.6.2.27 `static void ah_hmac_sha2_512_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 989 of file `ah_core.c`.

7.6.2.28 `static int ah_keyed_md5_init (struct ah_algorithm_state * state, struct secasvar * sav) [static]`

Definition at line 326 of file `ah_core.c`.

7.6.2.29 `static void ah_keyed_md5_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 384 of file `ah_core.c`.

7.6.2.30 `static int ah_keyed_md5_mature (struct secasvar * sav) [static]`

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

7.6.2.31 `static void ah_keyed_md5_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 396 of file `ah_core.c`.

References MD5_RESULTLEN.

7.6.2.32 `static int ah_keyed_sha1_init (struct ah_algorithm_state * state, struct secasvar * sav) [static]`

Definition at line 417 of file `ah_core.c`.

7.6.2.33 `static void ah_keyed_sha1_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len) [static]`

Definition at line 475 of file `ah_core.c`.

7.6.2.34 `static void ah_keyed_sha1_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l) [static]`

Definition at line 490 of file `ah_core.c`.

7.6.2.35 `static int ah_none_init (struct ah_algorithm_state * state, struct secasvar * sav)`
[static]

Definition at line 293 of file ah_core.c.

7.6.2.36 `static void ah_none_loop (struct ah_algorithm_state * state, u_int8_t * addr, size_t len)`
[static]

Definition at line 302 of file ah_core.c.

7.6.2.37 `static int ah_none_mature (struct secasvar * sav)` [static]

Definition at line 281 of file ah_core.c.

References ipseclog.

7.6.2.38 `static void ah_none_result (struct ah_algorithm_state * state, u_int8_t * addr, size_t l)`
[static]

Definition at line 310 of file ah_core.c.

7.6.2.39 `static int ah_sumsiz_1216 (struct secasvar * sav)` [static]

Definition at line 232 of file ah_core.c.

7.6.2.40 `static int ah_sumsiz_zero (struct secasvar * sav)` [static]

Definition at line 244 of file ah_core.c.

7.6.2.41 `static void ah_update_mbuf (struct mbuf * m, int off, int len, const struct ah_algorithm * algo, struct ah_algorithm_state * algos)` [static]

Definition at line 1129 of file ah_core.c.

7.6.3 Variable Documentation

7.6.3.1 `struct ah_algorithm ah_algorithms[]` [static]

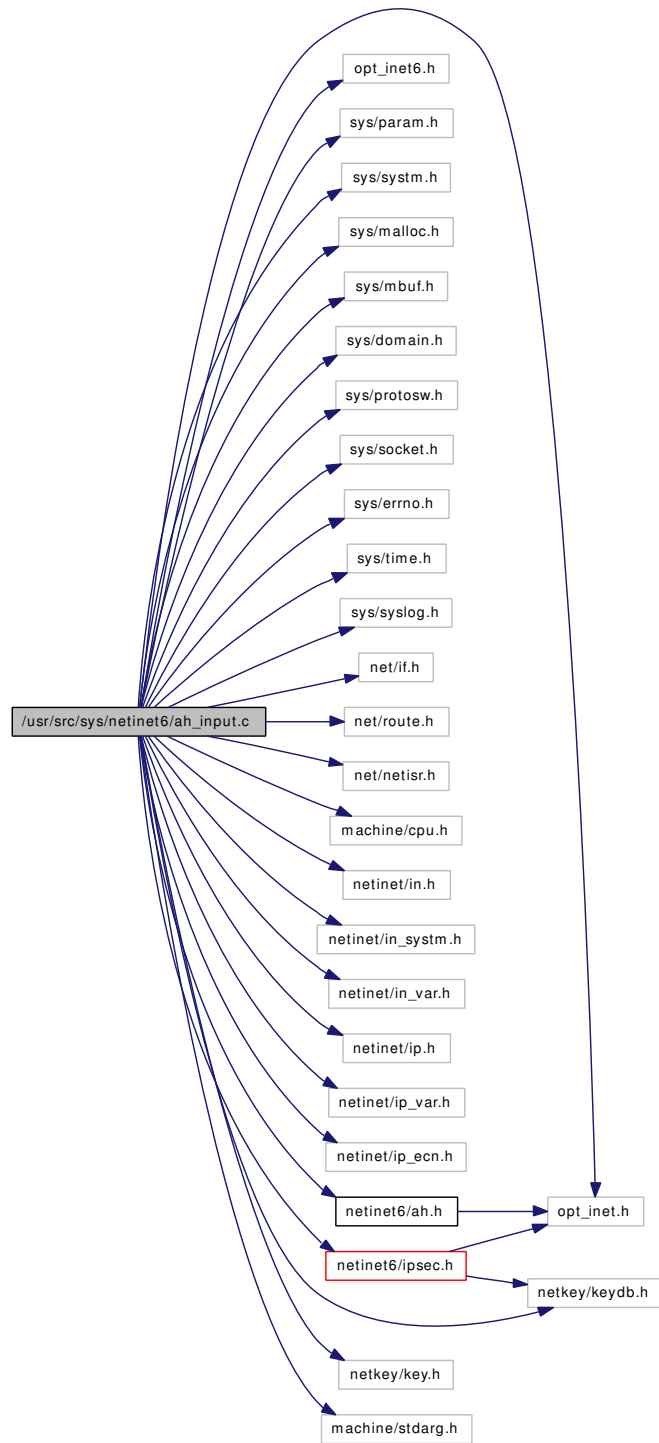
Definition at line 156 of file ah_core.c.

Referenced by ah_algorithm_lookup().

7.7 /usr/src/sys/netinet6/ah_input.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/malloc.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <net/netisr.h>  
#include <machine/cpu.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/in_var.h>  
#include <netinet/ip.h>  
#include <netinet/ip_var.h>  
#include <netinet/ip_ecn.h>  
#include <netinet6/ipsec.h>  
#include <netinet6/ah.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>  
#include <machine/stdarg.h>
```

Include dependency graph for ah_input.c:



Defines

- #define [KEYDEBUG](#)(lev, arg)
- #define [IPLen_FLIPPED](#)

7.7.1 Define Documentation

7.7.1.1 #define IPLEN_FLIPPED

Definition at line 93 of file ah_input.c.

7.7.1.2 #define KEYDEBUG(lev, arg)

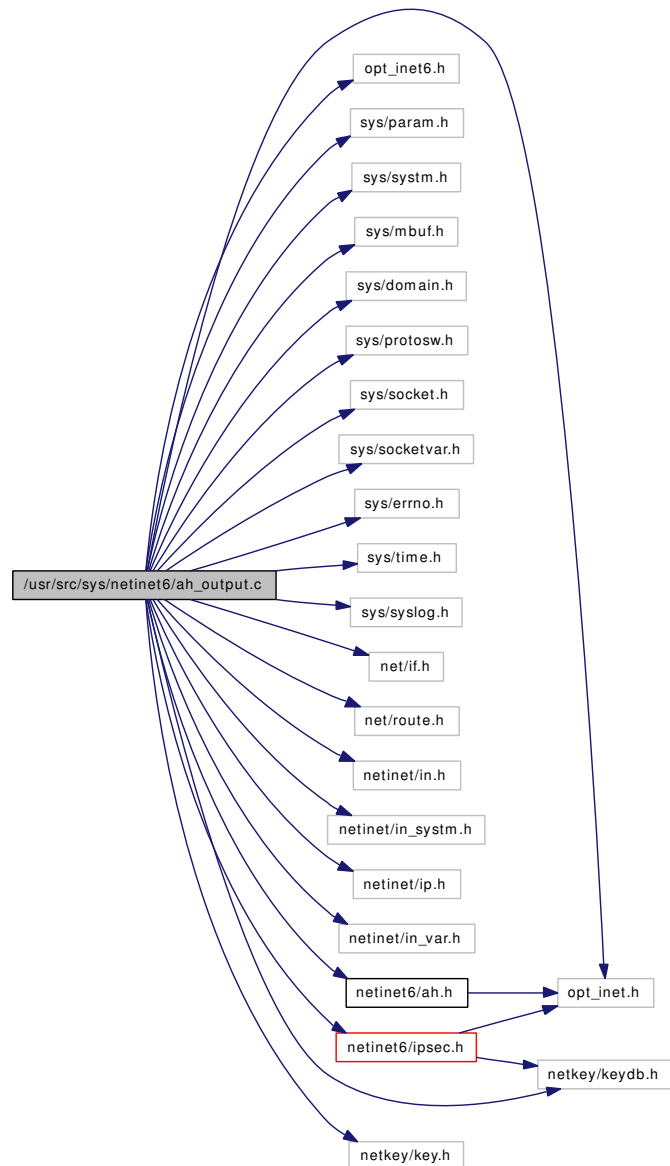
Definition at line 88 of file ah_input.c.

Referenced by ipsec4_getpolicybyaddr(), ipsec4_getpolicybypcb(), ipsec4_hdrsiz(), ipsec4_in_reject(), ipsec_checkpcbcache(), ipsec_fillpcbcache(), ipsec_get_policy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_set_policy(), and ipsec_setspidx().

7.8 /usr/src/sys/netinet6/ah_output.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/socketvar.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/ip.h>  
#include <netinet/in_var.h>  
#include <netinet6/ipsec.h>  
#include <netinet6/ah.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>
```

Include dependency graph for ah_output.c:



Functions

- `size_t ah_hdrsiz` (struct `ipsecrequest` *`isr`)
- `int ah_hdrlen` (struct `secasvar` *`sav`)

7.8.1 Function Documentation

7.8.1.1 `int ah_hdrlen` (struct `secasvar` * `sav`)

Definition at line 329 of file `ah_output.c`.

References `ah_algorithm_lookup()`.

Here is the call graph for this function:



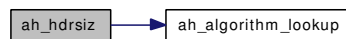
7.8.1.2 `size_t` ah_hdrsiz (struct ipsecrequest * isr)

Definition at line 87 of file ah_output.c.

References ah_algorithm_lookup(), and AH_MAXSUMSIZE.

Referenced by ipsec_hdrsiz().

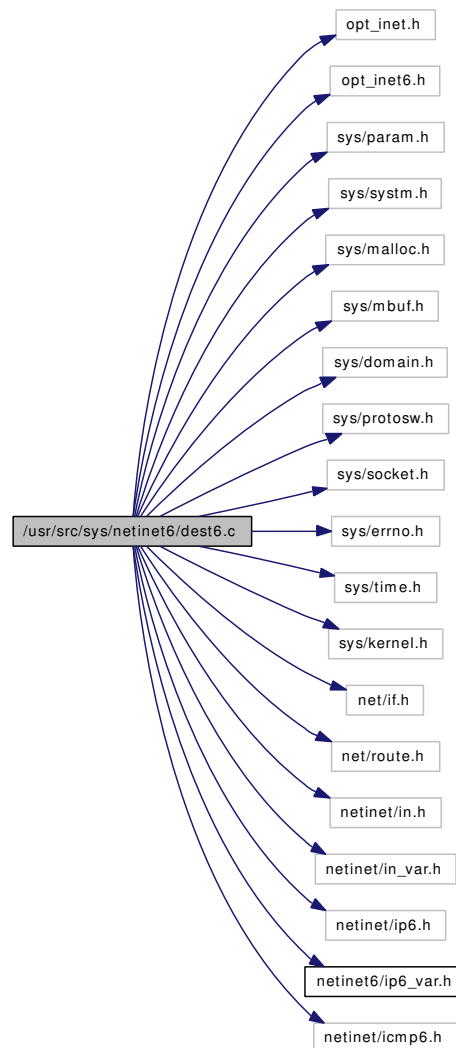
Here is the call graph for this function:



7.9 /usr/src/sys/netinet6/dest6.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/malloc.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/kernel.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_var.h>  
#include <netinet/ip6.h>  
#include <netinet6/ip6_var.h>  
#include <netinet/icmp6.h>
```

Include dependency graph for dest6.c:



Functions

- `int dest6_input` (struct mbuf **mp, int *offp, int proto)

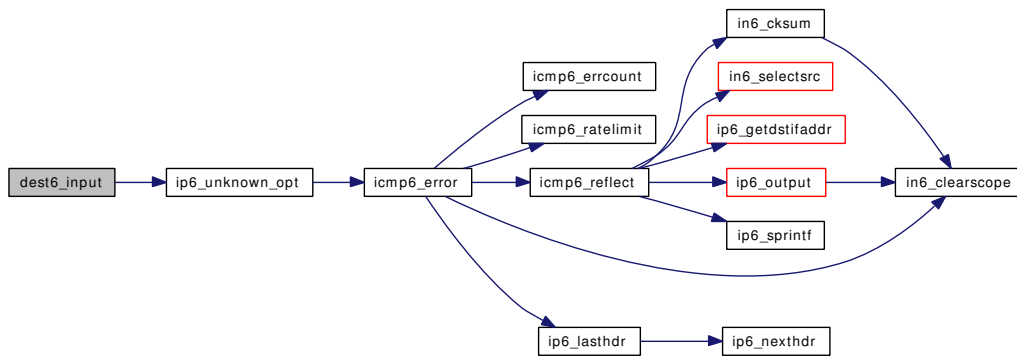
7.9.1 Function Documentation

7.9.1.1 `int dest6_input` (struct mbuf ** mp, int * offp, int proto)

Definition at line 60 of file dest6.c.

References `ip6_unknown_opt()`.

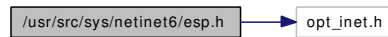
Here is the call graph for this function:



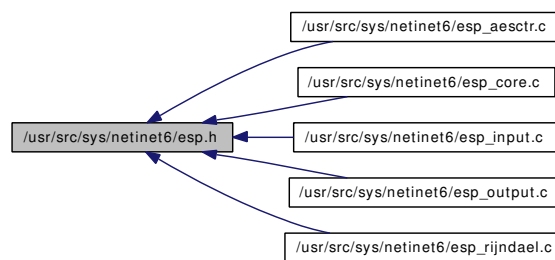
7.10 /usr/src/sys/netinet6/esp.h File Reference

```
#include "opt_inet.h"
```

Include dependency graph for esp.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [esp](#)
- struct [newesp](#)
- struct [esptail](#)
- struct [esp_algorithm](#)

Functions

- [esp_algorithm](#) *esp_algorithm_lookup __P ((int))
- int esp_max_ivlen __P ((void))
- int esp4_output __P ((struct mbuf *, struct [ipsecrequest](#) *))
- void esp4_input __P ((struct mbuf *, int))
- [size_t](#) esp_hdrsiz __P ((struct [ipsecrequest](#) *))
- int esp_schedule __P ((const struct [esp_algorithm](#) *, struct secasvar *))
- int esp_auth __P ((struct mbuf *, [size_t](#), [size_t](#), struct secasvar *, u_char *))

7.10.1 Function Documentation

7.10.1.1 `int esp_auth __P ((struct mbuf *, size_t, size_t, struct secasvar *, u_char *))`

7.10.1.2 `int esp_schedule __P ((const struct esp_algorithm *, struct secasvar *))`

7.10.1.3 `size_t esp_hdrsiz __P ((struct ipsecrequest *))`

7.10.1.4 `void esp4_input __P ((struct mbuf *, int))`

7.10.1.5 `int esp4_output __P ((struct mbuf *, struct ipsecrequest *))`

7.10.1.6 `void pfxlist_onlink_check __P ((void))`

7.10.1.7 `struct esp_algorithm* esp_algorithm_lookup __P ((int))`

7.11 /usr/src/sys/netinet6/esp6.h File Reference

Functions

- int esp6_output __P((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))
- int esp6_input __P((struct mbuf **, int *, int))
- void esp6_ctlinput __P((int, struct sockaddr *, void *))

7.11.1 Function Documentation

7.11.1.1 void esp6_ctlinput __P((int, struct sockaddr *, void *))

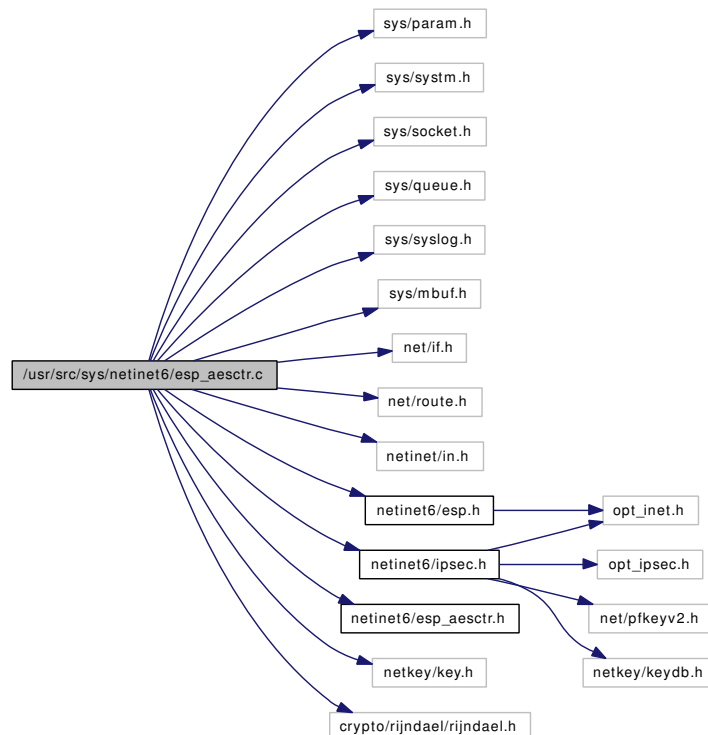
7.11.1.2 int esp6_input __P((struct mbuf **, int *, int))

7.11.1.3 int esp6_output __P((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))

7.12 /usr/src/sys/netinet6/esp_aesctr.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/socket.h>
#include <sys/queue.h>
#include <sys/syslog.h>
#include <sys/mbuf.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet6/ipsec.h>
#include <netinet6/esp.h>
#include <netinet6/esp_aesctr.h>
#include <netkey/key.h>
#include <crypto/rijndael/rijndael.h>
```

Include dependency graph for esp_aesctr.c:



Data Structures

- union [cblock](#)

- struct [aesctr_ctx](#)

Defines

- #define [AES_BLOCKSIZE](#) 16
- #define [NONCESIZE](#) 4

Functions

- int [esp_aesctr_mature](#) (struct [secasvar](#) *sav)
- [size_t](#) [esp_aesctr_schedlen](#) (struct [esp_algorithm](#) *algo) const
- int [esp_aesctr_schedule](#) (struct [esp_algorithm](#) *algo, struct [secasvar](#) *sav) const
- int [esp_aesctr_decrypt](#) (struct [mbuf](#) *m, [size_t](#) off, struct [secasvar](#) *sav, const struct [esp_algorithm](#) *algo, int ivlen)
- int [esp_aesctr_encrypt](#) (struct [mbuf](#) *m, [size_t](#) off, [size_t](#) plen, struct [secasvar](#) *sav, const struct [esp_algorithm](#) *algo, int ivlen)

7.12.1 Define Documentation

7.12.1.1 #define AES_BLOCKSIZE 16

Definition at line 54 of file [esp_aesctr.c](#).

7.12.1.2 #define NONCESIZE 4

Definition at line 56 of file [esp_aesctr.c](#).

Referenced by [esp_aesctr_decrypt\(\)](#), [esp_aesctr_encrypt\(\)](#), and [esp_aesctr_schedule\(\)](#).

7.12.2 Function Documentation

7.12.2.1 int esp_aesctr_decrypt (struct mbuf * m, size_t off, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen)

Definition at line 132 of file [esp_aesctr.c](#).

References [AES_BLOCKSIZE](#), [cblock::cblock](#), [cblock::ctr](#), [ipseclog](#), [cblock::iv](#), [esp_algorithm::name](#), [cblock::nonce](#), [NONCESIZE](#), [esp_algorithm::padbound](#), [aesctr_ctx::r_ek](#), [aesctr_ctx::r_nr](#), and [cblock::v](#).

7.12.2.2 int esp_aesctr_encrypt (struct mbuf * m, size_t off, size_t plen, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen)

Definition at line 298 of file [esp_aesctr.c](#).

References [AES_BLOCKSIZE](#), [cblock::cblock](#), [cblock::ctr](#), [ipseclog](#), [cblock::iv](#), [esp_algorithm::name](#), [cblock::nonce](#), [NONCESIZE](#), [esp_algorithm::padbound](#), [aesctr_ctx::r_ek](#), [aesctr_ctx::r_nr](#), and [cblock::v](#).

7.12.2.3 `int esp_aesctr_mature (struct secasvar * sav)`

Definition at line 72 of file esp_aesctr.c.

References `esp_algorithm_lookup()`, `ipseclog`, `esp_algorithm::keymax`, `esp_algorithm::keymin`, and `esp_algorithm::name`.

Here is the call graph for this function:



7.12.2.4 `size_t esp_aesctr_schedlen (struct esp_algorithm * algo) const`

Definition at line 106 of file esp_aesctr.c.

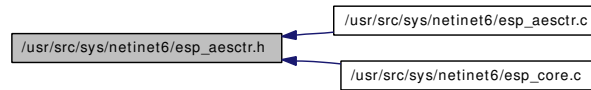
7.12.2.5 `int esp_aesctr_schedule (struct esp_algorithm * algo, struct secasvar * sav) const`

Definition at line 114 of file esp_aesctr.c.

References `NONCESIZE`, `aesctr_ctx::r_ek`, and `aesctr_ctx::r_nr`.

7.13 /usr/src/sys/netinet6/esp_aesctr.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int esp_aesctr_mature __P ((struct secasvar *))`
- `size_t esp_aesctr_schedlen __P ((const struct esp_algorithm *))`
- `int esp_aesctr_schedule __P ((const struct esp_algorithm *, struct secasvar *))`
- `int esp_aesctr_decrypt __P ((struct mbuf *, size_t, struct secasvar *, const struct esp_algorithm *, int))`
- `int esp_aesctr_encrypt __P ((struct mbuf *, size_t, size_t, struct secasvar *, const struct esp_algorithm *, int))`

7.13.1 Function Documentation

7.13.1.1 `static int esp_cbc_encrypt __P ((struct mbuf *, size_t, size_t, struct secasvar *, const struct esp_algorithm *, int))`

7.13.1.2 `static int esp_cbc_decrypt __P ((struct mbuf *, size_t, struct secasvar *, const struct esp_algorithm *, int))`

7.13.1.3 `static int esp_common_ivlen __P ((const struct esp_algorithm *, struct secasvar *))`

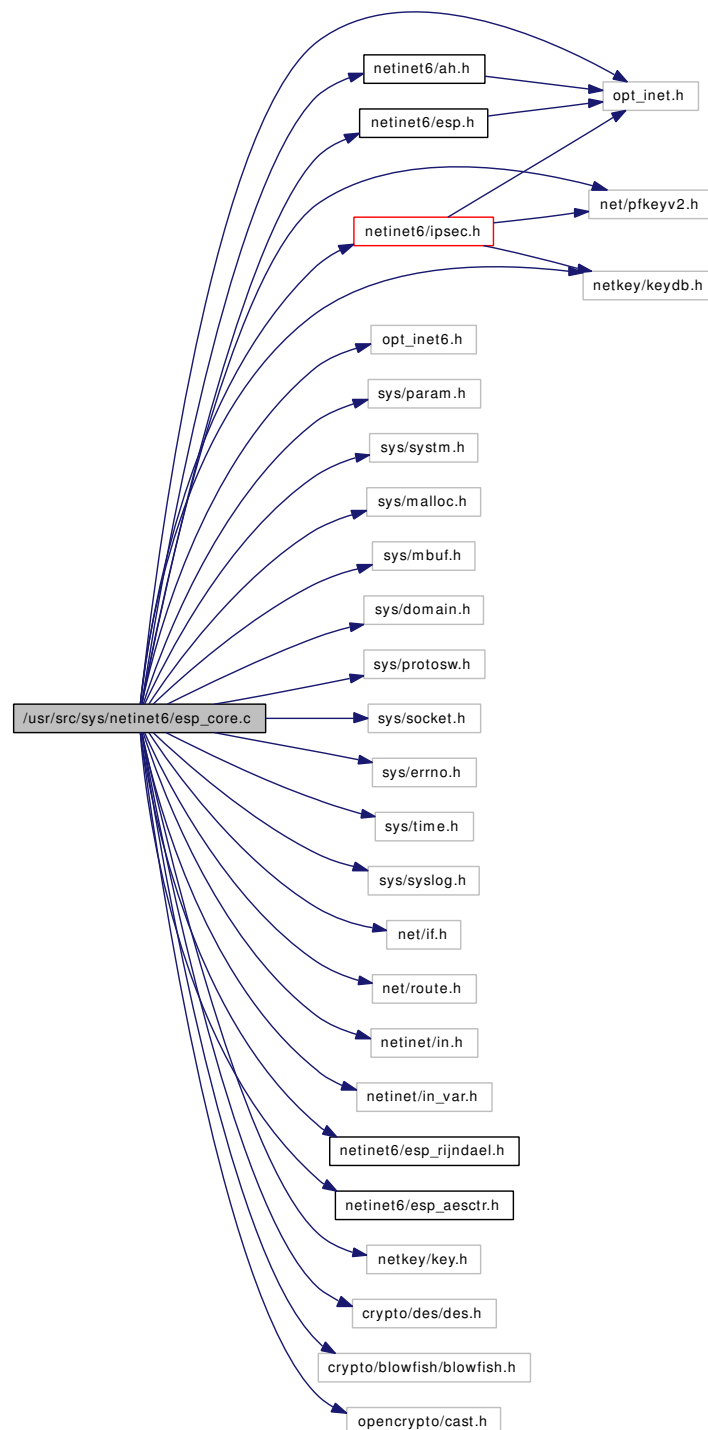
7.13.1.4 `static size_t esp_3des_schedlen __P ((const struct esp_algorithm *))`

7.13.1.5 `int esp_aesctr_mature __P ((struct secasvar *))`

7.14 /usr/src/sys/netinet6/esp_core.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/malloc.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_var.h>  
#include <netinet6/ipsec.h>  
#include <netinet6/ah.h>  
#include <netinet6/esp.h>  
#include <netinet6/esp_rijndael.h>  
#include <netinet6/esp_aesctr.h>  
#include <net/pfkeyv2.h>  
#include <netkey/keydb.h>  
#include <netkey/key.h>  
#include <crypto/des/des.h>  
#include <crypto/blowfish/blowfish.h>  
#include <opencrypto/cast.h>
```

Include dependency graph for esp_core.c:



Defines

- #define `cast128_key` `cast_key`
- #define `cast128_setkey`(key, rawkey, keybytes) `cast_setkey`((key), (rawkey), (keybytes))
- #define `cast128_encrypt`(key, inblock, outblock) `cast_encrypt`((key), (inblock), (outblock))

- #define `cast128_decrypt`(key, inblock, outblock) `cast_decrypt`((key), (inblock), (outblock))
- #define `MAXIVLEN` 16

Functions

- static int `esp_null_mature` `__P` ((struct `secasvar` *)
- static int `esp_null_decrypt` `__P` ((struct `mbuf` *, `size_t`, struct `secasvar` *, const struct `esp_algorithm` *, int))
- static int `esp_null_encrypt` `__P` ((struct `mbuf` *, `size_t`, `size_t`, struct `secasvar` *, const struct `esp_algorithm` *, int))
- static int `esp_descbc_ivlen` `__P` ((const struct `esp_algorithm` *, struct `secasvar` *)
- static `size_t` `esp_des_schedlen` `__P` ((const struct `esp_algorithm` *)
- static int `esp_des_blockdecrypt` `__P` ((const struct `esp_algorithm` *, struct `secasvar` *, `u_int8_t` *, `u_int8_t` *)
- `esp_algorithm` * `esp_algorithm_lookup` (int `idx`)
- int `esp_max_ivlen` ()
- int `esp_schedule` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static int `esp_null_mature` (struct `secasvar` *`sav`)
- static int `esp_null_decrypt` (struct `mbuf` *`m`, `size_t` `off`, struct `secasvar` *`sav`, const struct `esp_algorithm` *`algo`, int `ivlen`)
- static int `esp_null_encrypt` (struct `mbuf` *`m`, `size_t` `off`, `size_t` `plen`, struct `secasvar` *`sav`, const struct `esp_algorithm` *`algo`, int `ivlen`)
- static int `esp_descbc_mature` (struct `secasvar` *`sav`)
- static int `esp_descbc_ivlen` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static `size_t` `esp_des_schedlen` (struct `esp_algorithm` *`algo`) const
- static int `esp_des_schedule` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static int `esp_des_blockdecrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_des_blockencrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_cbc_mature` (struct `secasvar` *`sav`)
- static `size_t` `esp_blowfish_schedlen` (struct `esp_algorithm` *`algo`) const
- static int `esp_blowfish_schedule` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static int `esp_blowfish_blockdecrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_blowfish_blockencrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static `size_t` `esp_cast128_schedlen` (struct `esp_algorithm` *`algo`) const
- static int `esp_cast128_schedule` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static int `esp_cast128_blockdecrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_cast128_blockencrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static `size_t` `esp_3des_schedlen` (struct `esp_algorithm` *`algo`) const
- static int `esp_3des_schedule` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const
- static int `esp_3des_blockdecrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_3des_blockencrypt` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`, `u_int8_t` *`s`, `u_int8_t` *`d`) const
- static int `esp_common_ivlen` (struct `esp_algorithm` *`algo`, struct `secasvar` *`sav`) const

- static int `esp_cbc_decrypt` (struct mbuf *m, `size_t` off, struct secasvar *sav, const struct `esp_algorithm` *algo, int ivlen)
- static int `esp_cbc_encrypt` (struct mbuf *m, `size_t` off, `size_t` plen, struct secasvar *sav, const struct `esp_algorithm` *algo, int ivlen)
- int `esp_auth` (struct mbuf *m0, `size_t` skip, `size_t` length, struct secasvar *sav, u_char *sum)

Variables

- static struct `esp_algorithm` `esp_algorithms` []

7.14.1 Define Documentation

7.14.1.1 `#define cast128_decrypt(key, inblock, outblock) cast_decrypt((key), (inblock), (outblock))`

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

Referenced by `esp_cast128_blockdecrypt()`.

7.14.1.2 `#define cast128_encrypt(key, inblock, outblock) cast_encrypt((key), (inblock), (outblock))`

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

Referenced by `esp_cast128_blockencrypt()`.

7.14.1.3 `#define cast128_key cast_key`

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

Referenced by `esp_cast128_blockdecrypt()`, `esp_cast128_blockencrypt()`, `esp_cast128_schedlen()`, and `esp_cast128_schedule()`.

7.14.1.4 `#define cast128_setkey(key, rawkey, keybytes) cast_setkey((key), (rawkey), (keybytes))`

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

Referenced by `esp_cast128_schedule()`.

7.14.1.5 `#define MAXIVLEN 16`

Definition at line 132 of file `esp_core.c`.

Referenced by `esp_cbc_decrypt()`, and `esp_cbc_encrypt()`.

7.14.2 Function Documentation

7.14.2.1 `static int esp_3des_blockencrypt __P ((const struct esp_algorithm *, struct secasvar *, u_int8_t *, u_int8_t *))` [static]

7.14.2.2 `static size_t esp_des_schedlen __P ((const struct esp_algorithm *))` [static]

7.14.2.3 `static int esp_descbc_ivlen __P ((const struct esp_algorithm *, struct secasvar *))` [static]

7.14.2.4 `static int esp_null_encrypt __P ((struct mbuf *, size_t, size_t, struct secasvar *, const struct esp_algorithm *, int))` [static]

7.14.2.5 `static int esp_null_decrypt __P ((struct mbuf *, size_t, struct secasvar *, const struct esp_algorithm *, int))` [static]

7.14.2.6 `static int esp_null_mature __P ((struct secasvar *))` [static]

7.14.2.7 `static int esp_3des_blockdecrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 573 of file esp_core.c.

7.14.2.8 `static int esp_3des_blockencrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 590 of file esp_core.c.

7.14.2.9 `static size_t esp_3des_schedlen (struct esp_algorithm * algo) const` [static]

Definition at line 545 of file esp_core.c.

7.14.2.10 `static int esp_3des_schedule (struct esp_algorithm * algo, struct secasvar * sav) const` [static]

Definition at line 553 of file esp_core.c.

7.14.2.11 `struct esp_algorithm* esp_algorithm_lookup (int idx)`

Definition at line 168 of file esp_core.c.

References esp_algorithms.

Referenced by esp_aesctr_mature(), esp_cbc_mature(), esp_descbc_mature(), esp_hdrsiz(), and esp_output().

7.14.2.12 `int esp_auth (struct mbuf * m0, size_t skip, size_t length, struct secasvar * sav, u_char * sum)`

Definition at line 1032 of file esp_core.c.

References `ah_algorithm_lookup()`, `AH_MAXSUMSIZE`, and `ipseclog`.

Here is the call graph for this function:



7.14.2.13 `static int esp_blowfish_blockdecrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 478 of file `esp_core.c`.

7.14.2.14 `static int esp_blowfish_blockencrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 490 of file `esp_core.c`.

7.14.2.15 `static size_t esp_blowfish_schedlen (struct esp_algorithm * algo) const` [static]

Definition at line 459 of file `esp_core.c`.

7.14.2.16 `static int esp_blowfish_schedule (struct esp_algorithm * algo, struct secasvar * sav) const` [static]

Definition at line 467 of file `esp_core.c`.

7.14.2.17 `static int esp_cast128_blockdecrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 521 of file `esp_core.c`.

References `cast128_decrypt`, and `cast128_key`.

7.14.2.18 `static int esp_cast128_blockencrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const` [static]

Definition at line 533 of file `esp_core.c`.

References `cast128_encrypt`, and `cast128_key`.

7.14.2.19 `static size_t esp_cast128_schedlen (struct esp_algorithm * algo) const` [static]

Definition at line 502 of file `esp_core.c`.

References `cast128_key`.

7.14.2.20 `static int esp_cast128_schedule (struct esp_algorithm * algo, struct secasvar * sav) const [static]`

Definition at line 510 of file esp_core.c.

References cast128_key, and cast128_setkey.

7.14.2.21 `static int esp_cbc_decrypt (struct mbuf * m, size_t off, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen) [static]`

Definition at line 618 of file esp_core.c.

References ipseclog, MAXIVLEN, esp_algorithm::name, and esp_algorithm::padbound.

7.14.2.22 `static int esp_cbc_encrypt (struct mbuf * m, size_t off, size_t plen, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen) [static]`

Definition at line 821 of file esp_core.c.

References ipseclog, MAXIVLEN, esp_algorithm::name, and esp_algorithm::padbound.

7.14.2.23 `static int esp_cbc_mature (struct secasvar * sav) [static]`

Definition at line 393 of file esp_core.c.

References esp_algorithm_lookup(), ipseclog, esp_algorithm::keymax, esp_algorithm::keymin, and esp_algorithm::name.

Here is the call graph for this function:



7.14.2.24 `static int esp_common_ivlen (struct esp_algorithm * algo, struct secasvar * sav) const [static]`

Definition at line 607 of file esp_core.c.

7.14.2.25 `static int esp_des_blockdecrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const [static]`

Definition at line 363 of file esp_core.c.

7.14.2.26 `static int esp_des_blockencrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const [static]`

Definition at line 378 of file esp_core.c.

7.14.2.27 `static size_t esp_des_schedlen (struct esp_algorithm * algo) const` [static]

Definition at line 342 of file esp_core.c.

7.14.2.28 `static int esp_des_schedule (struct esp_algorithm * algo, struct secasvar * sav) const` [static]

Definition at line 350 of file esp_core.c.

7.14.2.29 `static int esp_descbc_ivlen (struct esp_algorithm * algo, struct secasvar * sav) const` [static]

Definition at line 327 of file esp_core.c.

7.14.2.30 `static int esp_descbc_mature (struct secasvar * sav)` [static]

Definition at line 285 of file esp_core.c.

References esp_algorithm_lookup(), ipseclog, esp_algorithm::keymax, and esp_algorithm::keymin.

Here is the call graph for this function:



7.14.2.31 `int esp_max_ivlen ()`

Definition at line 193 of file esp_core.c.

References esp_algorithms, and esp_algorithm::ivlenval.

Referenced by esp_hdrsiz().

7.14.2.32 `static int esp_null_decrypt (struct mbuf * m, size_t off, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen)` [static]

Definition at line 260 of file esp_core.c.

7.14.2.33 `static int esp_null_encrypt (struct mbuf * m, size_t off, size_t plen, struct secasvar * sav, const struct esp_algorithm * algo, int ivlen)` [static]

Definition at line 272 of file esp_core.c.

7.14.2.34 `static int esp_null_mature (struct secasvar * sav)` [static]

Definition at line 251 of file esp_core.c.

7.14.2.35 `int esp_schedule (struct esp_algorithm * algo, struct secasvar * sav) const`

Definition at line 208 of file `esp_core.c`.

References `ipsecclog`.

7.14.3 Variable Documentation

7.14.3.1 `struct esp_algorithm esp_algorithms[] [static]`

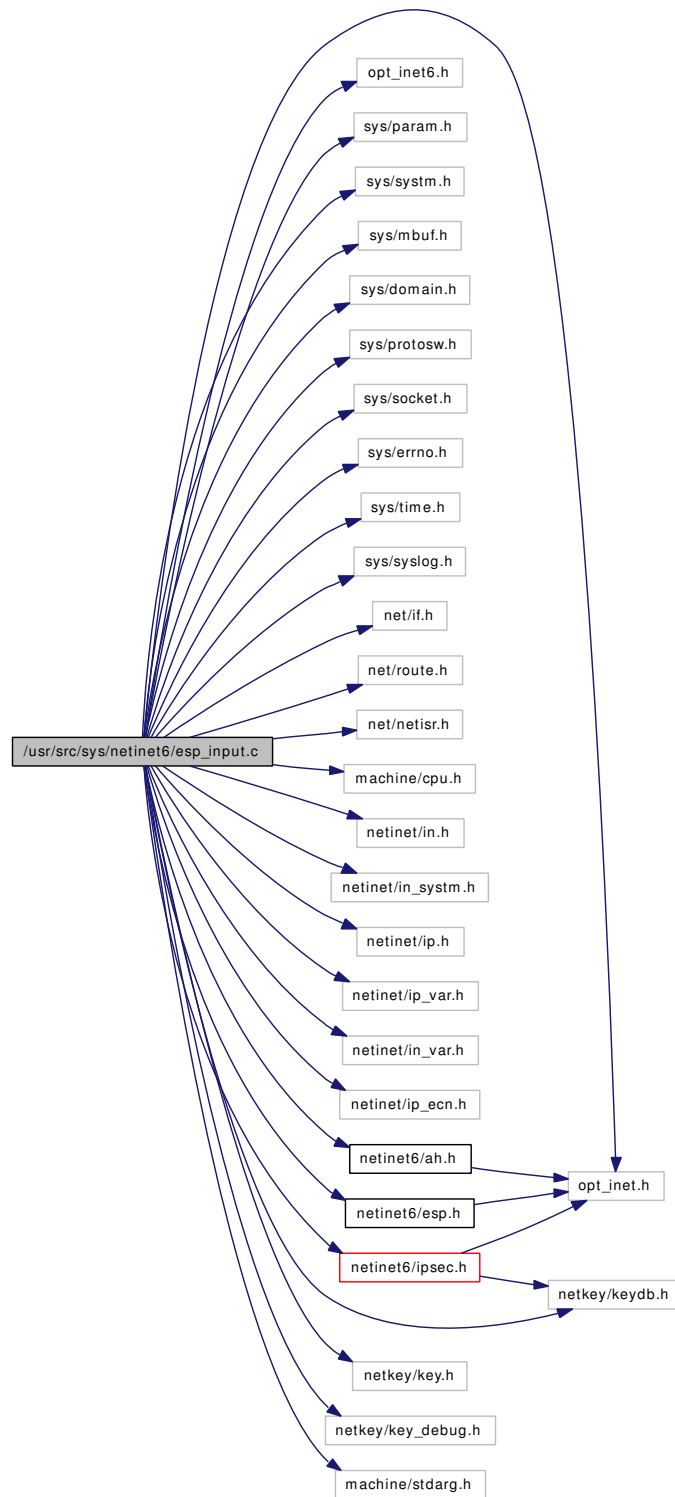
Definition at line 134 of file `esp_core.c`.

Referenced by `esp_algorithm_lookup()`, and `esp_max_ivlen()`.

7.15 /usr/src/sys/netinet6/esp_input.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <net/netisr.h>  
#include <machine/cpu.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/ip.h>  
#include <netinet/ip_var.h>  
#include <netinet/in_var.h>  
#include <netinet/ip_ecn.h>  
#include <netinet6/ipsec.h>  
#include <netinet6/ah.h>  
#include <netinet6/esp.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>  
#include <netkey/key_debug.h>  
#include <machine/stdarg.h>
```

Include dependency graph for esp_input.c:



Defines

- `#define IPLEN_FLIPPED`

- #define [ESP_MAXLEN](#)

7.15.1 Define Documentation

7.15.1.1 #define ESP_MAXLEN

Value:

```
(sizeof(struct esp) < sizeof(struct newesp) \
    ? sizeof(struct newesp) : sizeof(struct esp))
```

Definition at line 94 of file esp_input.c.

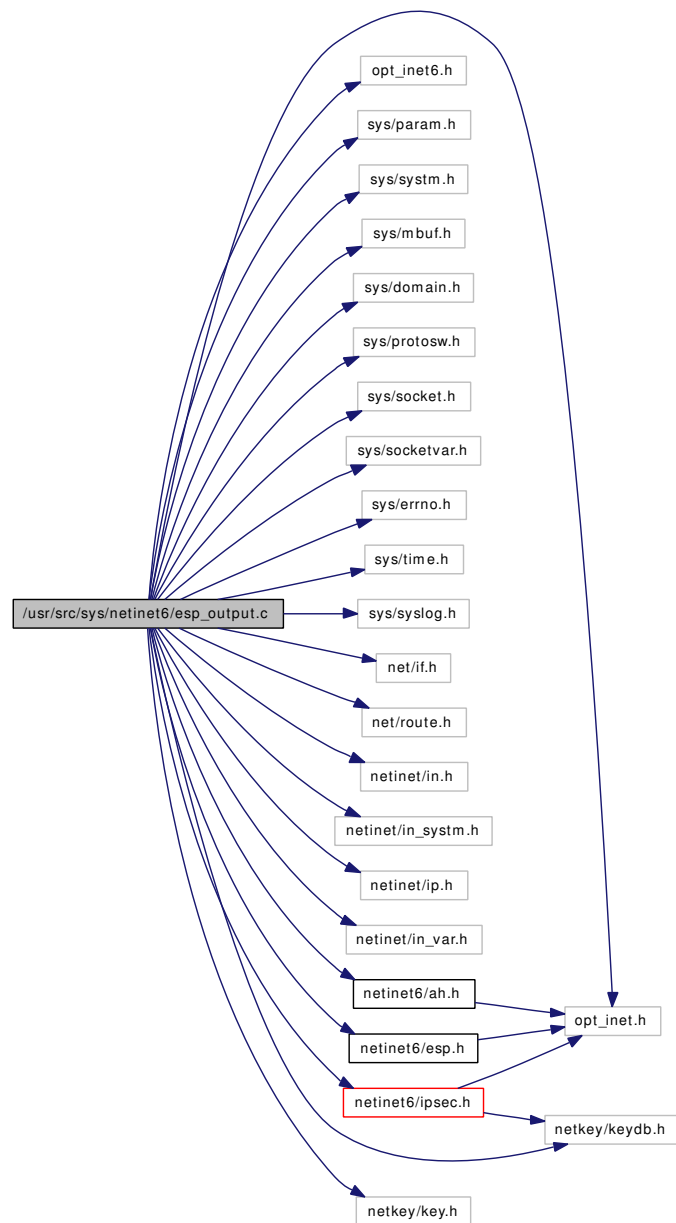
7.15.1.2 #define IPLEN_FLIPPED

Definition at line 92 of file esp_input.c.

7.16 /usr/src/sys/netinet6/esp_output.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/socketvar.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/ip.h>  
#include <netinet/in_var.h>  
#include <netinet6/ipsec.h>  
#include <netinet6/ah.h>  
#include <netinet6/esp.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>
```

Include dependency graph for esp_output.c:



Functions

- static int esp_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *, int))
- size_t esp_hdrsiz (struct ipsecrequest *isr)
- static int esp_output (struct mbuf *m, u_char *nexthdrp, struct mbuf *md, struct ipsecrequest *isr, int af)

7.16.1 Function Documentation

7.16.1.1 `static int esp_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *, int))` `[static]`

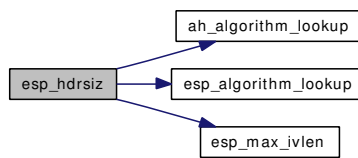
7.16.1.2 `size_t esp_hdrsiz (struct ipsecrequest * isr)`

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

References `ah_algorithm_lookup()`, `esp_algorithm_lookup()`, and `esp_max_ivlen()`.

Referenced by `ipsec_hdrsiz()`.

Here is the call graph for this function:

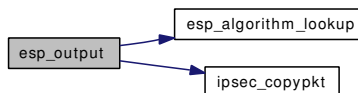


7.16.1.3 `static int esp_output (struct mbuf * m, u_char * nexthdrp, struct mbuf * md, struct ipsecrequest * isr, int af)` `[static]`

Definition at line 175 of file `esp_output.c`.

References `esp_algorithm_lookup()`, `ipsec6stat`, `ipsec_copypkt()`, `ipseclog`, `ipsecstat::out_inval`, and `ipsecrequest::sav`.

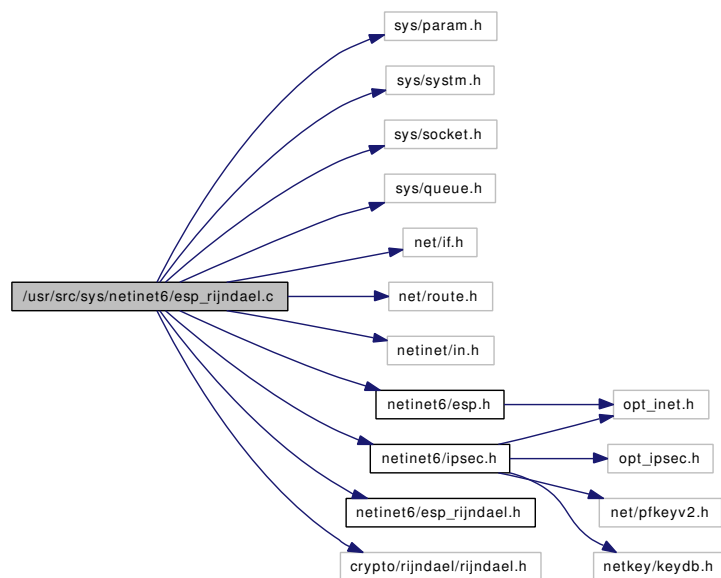
Here is the call graph for this function:



7.17 /usr/src/sys/netinet6/esp_rijndael.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/socket.h>
#include <sys/queue.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet6/ipsec.h>
#include <netinet6/esp.h>
#include <netinet6/esp_rijndael.h>
#include <crypto/rijndael/rijndael.h>
```

Include dependency graph for esp_rijndael.c:



Functions

- [size_t esp_rijndael_schedlen](#) (struct [esp_algorithm](#) *algo) const
- [int esp_rijndael_schedule](#) (struct [esp_algorithm](#) *algo, struct [secasvar](#) *sav) const
- [int esp_rijndael_blockdecrypt](#) (struct [esp_algorithm](#) *algo, struct [secasvar](#) *sav, u_int8_t *s, u_int8_t *d) const
- [int esp_rijndael_blockencrypt](#) (struct [esp_algorithm](#) *algo, struct [secasvar](#) *sav, u_int8_t *s, u_int8_t *d) const

7.17.1 Function Documentation

7.17.1.1 `int esp_rijndael_blockdecrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const`

Definition at line 70 of file esp_rijndael.c.

7.17.1.2 `int esp_rijndael_blockencrypt (struct esp_algorithm * algo, struct secasvar * sav, u_int8_t * s, u_int8_t * d) const`

Definition at line 84 of file esp_rijndael.c.

7.17.1.3 `size_t esp_rijndael_schedlen (struct esp_algorithm * algo) const`

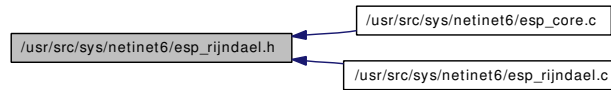
Definition at line 49 of file esp_rijndael.c.

7.17.1.4 `int esp_rijndael_schedule (struct esp_algorithm * algo, struct secasvar * sav) const`

Definition at line 57 of file esp_rijndael.c.

7.18 /usr/src/sys/netinet6/esp_rijndael.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `size_t esp_rijndael_schedlen __P ((const struct esp_algorithm *))`
- `int esp_rijndael_schedule __P ((const struct esp_algorithm *, struct secasvar *))`
- `int esp_rijndael_blockdecrypt __P ((const struct esp_algorithm *, struct secasvar *, u_int8_t *, u_int8_t *))`

7.18.1 Function Documentation

7.18.1.1 `int esp_rijndael_blockencrypt __P ((const struct esp_algorithm *, struct secasvar *, u_int8_t *, u_int8_t *))`

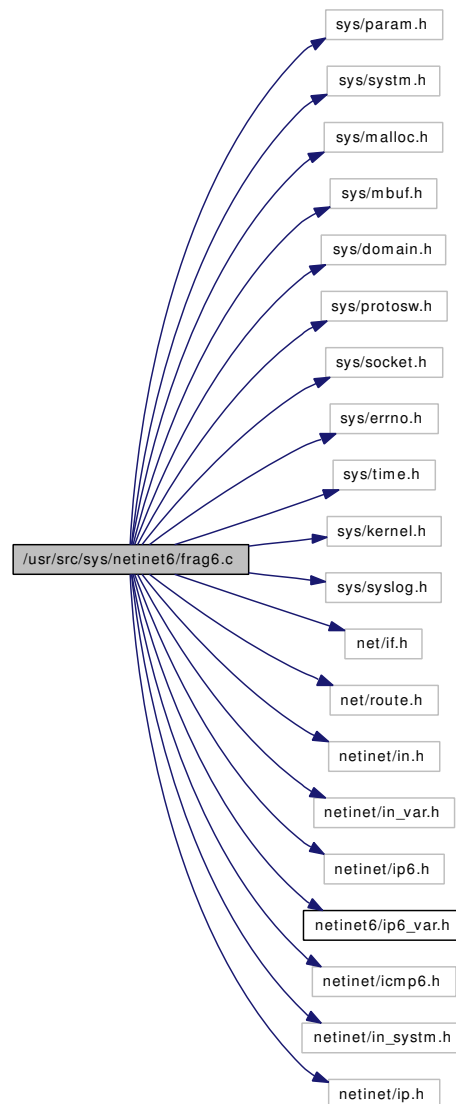
7.18.1.2 `int esp_rijndael_schedule __P ((const struct esp_algorithm *, struct secasvar *))`

7.18.1.3 `size_t esp_rijndael_schedlen __P ((const struct esp_algorithm *))`

7.19 /usr/src/sys/netinet6/frag6.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet/icmp6.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
```

Include dependency graph for frag6.c:



Defines

- `#define IN6_IFSTAT_STRICT`
- `#define IP6Q_LOCK_INIT() mtx_init(&ip6qlock, "ip6qlock", NULL, MTX_DEF);`
- `#define IP6Q_LOCK() mtx_lock(&ip6qlock)`
- `#define IP6Q_TRYLOCK() mtx_trylock(&ip6qlock)`
- `#define IP6Q_LOCK_ASSERT() mtx_assert(&ip6qlock, MA_OWNED)`
- `#define IP6Q_UNLOCK() mtx_unlock(&ip6qlock)`

Functions

- `static void frag6_enqueue __P ((struct ip6asfrag *, struct ip6asfrag *))`
- `static void frag6_dequeue __P ((struct ip6asfrag *))`
- `static void frag6_enqueue __P ((struct ip6q *, struct ip6q *))`
- `static void frag6_remove __P ((struct ip6q *))`

- static `MALLOC_DEFINE` (M_FTABLE,"fragment","fragment reassembly header")
- static void `frag6_change` (void *tag)
- void `frag6_init` ()
- int `frag6_input` (struct mbuf **mp, int *offp, int proto)
- void `frag6_freef` (struct `ip6q` *q6)
- void `frag6_enq` (struct `ip6asfrag` *af6, struct `ip6asfrag` *up6)
- void `frag6_deq` (struct `ip6asfrag` *af6)
- void `frag6_insque` (struct `ip6q` *new, struct `ip6q` *old)
- void `frag6_remque` (struct `ip6q` *p6)
- void `frag6_slowtimo` ()
- void `frag6_drain` ()

Variables

- static struct mtx `ip6qlock`
- static u_int `frag6_nfragpackets`
- static u_int `frag6_nfrags`
- static struct `ip6q` `ip6q`

7.19.1 Define Documentation

7.19.1.1 #define IN6_IFSTAT_STRICT

Definition at line 61 of file `frag6.c`.

7.19.1.2 #define IP6Q_LOCK() mtx_lock(&ip6qlock)

Definition at line 78 of file `frag6.c`.

Referenced by `frag6_input()`, and `frag6_slowtimo()`.

7.19.1.3 #define IP6Q_LOCK_ASSERT() mtx_assert(&ip6qlock, MA_OWNED)

Definition at line 80 of file `frag6.c`.

Referenced by `frag6_deq()`, `frag6_enq()`, `frag6_freef()`, `frag6_insque()`, and `frag6_remque()`.

7.19.1.4 #define IP6Q_LOCK_INIT() mtx_init(&ip6qlock, "ip6qlock", NULL, MTX_DEF);

Definition at line 77 of file `frag6.c`.

Referenced by `frag6_init()`.

7.19.1.5 #define IP6Q_TRYLOCK() mtx_trylock(&ip6qlock)

Definition at line 79 of file `frag6.c`.

Referenced by `frag6_drain()`.

7.19.1.6 #define IP6Q_UNLOCK() mtx_unlock(&ip6qlock)

Definition at line 81 of file frag6.c.

Referenced by frag6_drain(), frag6_input(), and frag6_slowtimo().

7.19.2 Function Documentation

7.19.2.1 static void frag6_freef __P ((struct ip6q *)) [static]

7.19.2.2 static void frag6_insqe __P ((struct ip6q *, struct ip6q *)) [static]

7.19.2.3 static void frag6_deq __P ((struct ip6asfrag *)) [static]

7.19.2.4 static void frag6_enq __P ((struct ip6asfrag *, struct ip6asfrag *)) [static]

7.19.2.5 static void frag6_change (void * tag) [static]

Definition at line 89 of file frag6.c.

References ip6_maxfragpackets, and ip6_maxfrags.

Referenced by frag6_init().

7.19.2.6 void frag6_deq (struct ip6asfrag * af6)

Definition at line 648 of file frag6.c.

References IP6Q_LOCK_ASSERT.

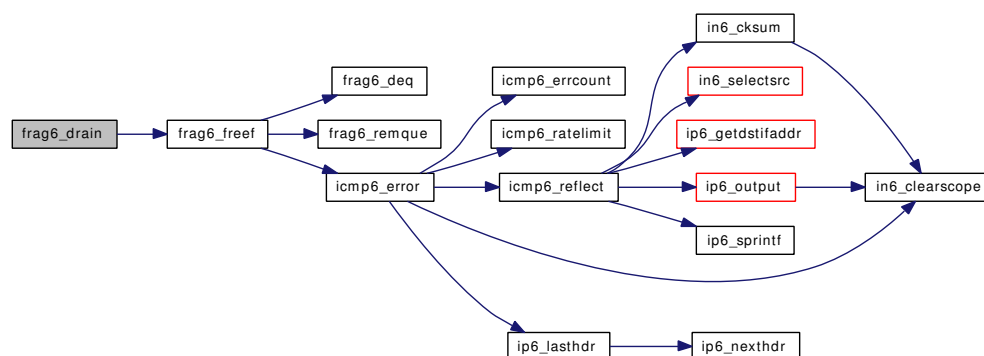
Referenced by frag6_freef(), and frag6_input().

7.19.2.7 void frag6_drain ()

Definition at line 742 of file frag6.c.

References frag6_freef(), ip6q, ip6q::ip6q_next, IP6Q_TRYLOCK, and IP6Q_UNLOCK.

Here is the call graph for this function:



7.19.2.8 void frag6_enq (struct ip6asfrag * af6, struct ip6asfrag * up6)

Definition at line 632 of file frag6.c.

References IP6Q_LOCK_ASSERT.

Referenced by frag6_input().

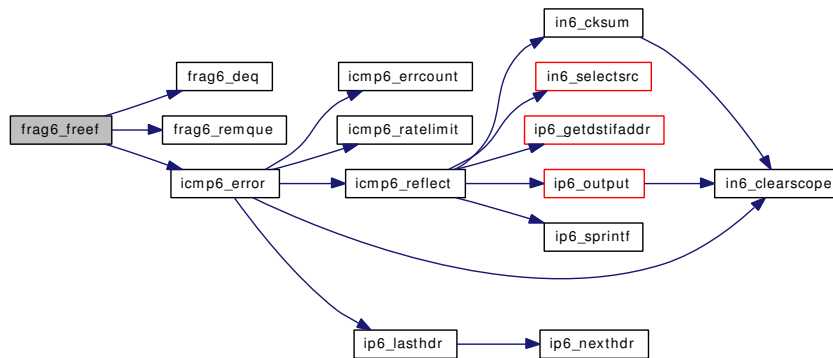
7.19.2.9 void frag6_freef (struct ip6q * q6)

Definition at line 587 of file frag6.c.

References frag6_deq(), frag6_remque(), icmp6_error(), IP6_REASS_MBUF, ip6asfrag::ip6af_down, ip6asfrag::ip6af_off, and IP6Q_LOCK_ASSERT.

Referenced by frag6_drain(), and frag6_slowtimo().

Here is the call graph for this function:



7.19.2.10 void frag6_init ()

Definition at line 97 of file frag6.c.

References frag6_change(), ip6_maxfragpackets, ip6_maxfrags, ip6q, IP6Q_LOCK_INIT, ip6q::ip6q_next, and ip6q::ip6q_prev.

Here is the call graph for this function:



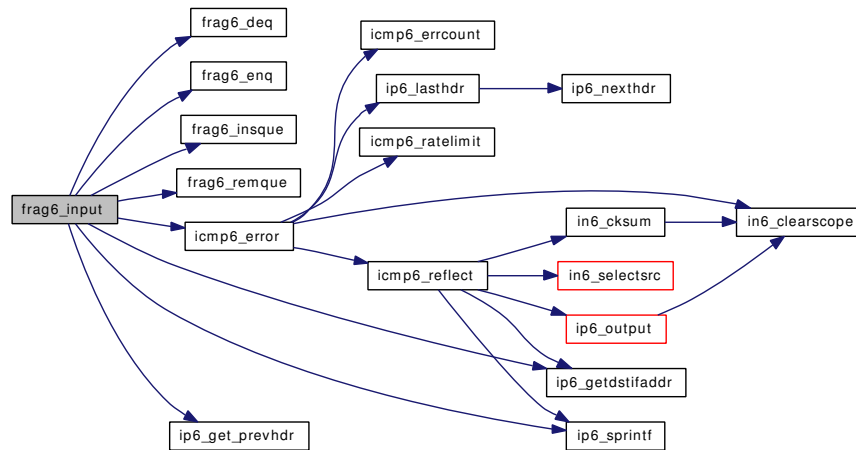
7.19.2.11 int frag6_input (struct mbuf ** mp, int * offp, int proto)

Definition at line 143 of file frag6.c.

References frag6_deq(), frag6_enq(), frag6_insque(), frag6_remque(), icmp6_error(), IN6_ADDR_EQUAL, in6_ifstat_inc, INET6_ADDRSTRLEN, ip6_get_prevhdr(), ip6_getdstifaddr(), ip6_maxfragpackets, ip6_maxfrags, IP6_REASS_MBUF, ip6_sprintf(), ip6asfrag::ip6af_down, ip6asfrag::ip6af_frglen, ip6asfrag::ip6af_head, ip6asfrag::ip6af_mff, ip6asfrag::ip6af_off,

ip6asfrag::ip6af_offset, ip6asfrag::ip6af_up, ip6q, ip6q::ip6q_down, ip6q::ip6q_dst, ip6q::ip6q_ident, IP6Q_LOCK, ip6q::ip6q_next, ip6q::ip6q_nxt, ip6q::ip6q_unfrglen, and IP6Q_UNLOCK.

Here is the call graph for this function:



7.19.2.12 void frag6_insque (struct ip6q * new, struct ip6q * old)

Definition at line 659 of file frag6.c.

References IP6Q_LOCK_ASSERT.

Referenced by frag6_input().

7.19.2.13 void frag6_remque (struct ip6q * p6)

Definition at line 672 of file frag6.c.

References IP6Q_LOCK_ASSERT.

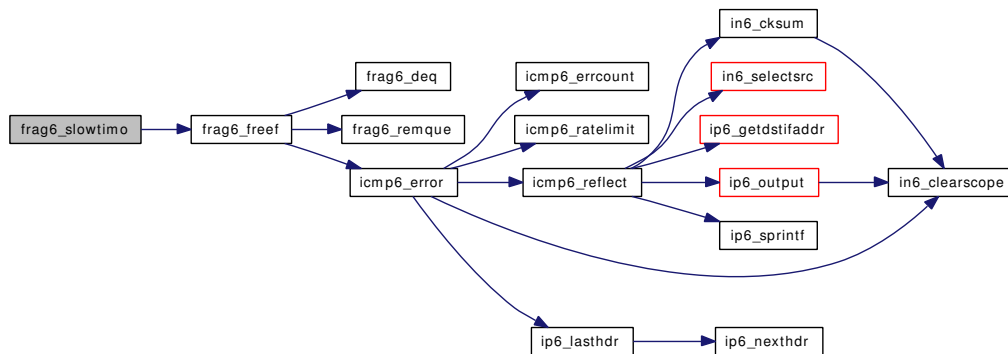
Referenced by frag6_free(), and frag6_input().

7.19.2.14 void frag6_slowtimo ()

Definition at line 688 of file frag6.c.

References frag6_free(), ip6_forward_rt, ip6_maxfragpackets, ip6q, IP6Q_LOCK, ip6q::ip6q_next, ip6q::ip6q_prev, ip6q::ip6q_ttl, and IP6Q_UNLOCK.

Here is the call graph for this function:



7.19.2.15 `static MALLOC_DEFINE (M_FTABLE, "fragment", "fragment reassembly header")`
`[static]`

7.19.3 Variable Documentation

7.19.3.1 `u_int frag6_nfragpackets` `[static]`

Definition at line 73 of file frag6.c.

7.19.3.2 `u_int frag6_nfrags` `[static]`

Definition at line 74 of file frag6.c.

7.19.3.3 `struct ip6q ip6q` `[static]`

Definition at line 75 of file frag6.c.

Referenced by frag6_drain(), frag6_init(), frag6_input(), and frag6_slowtimo().

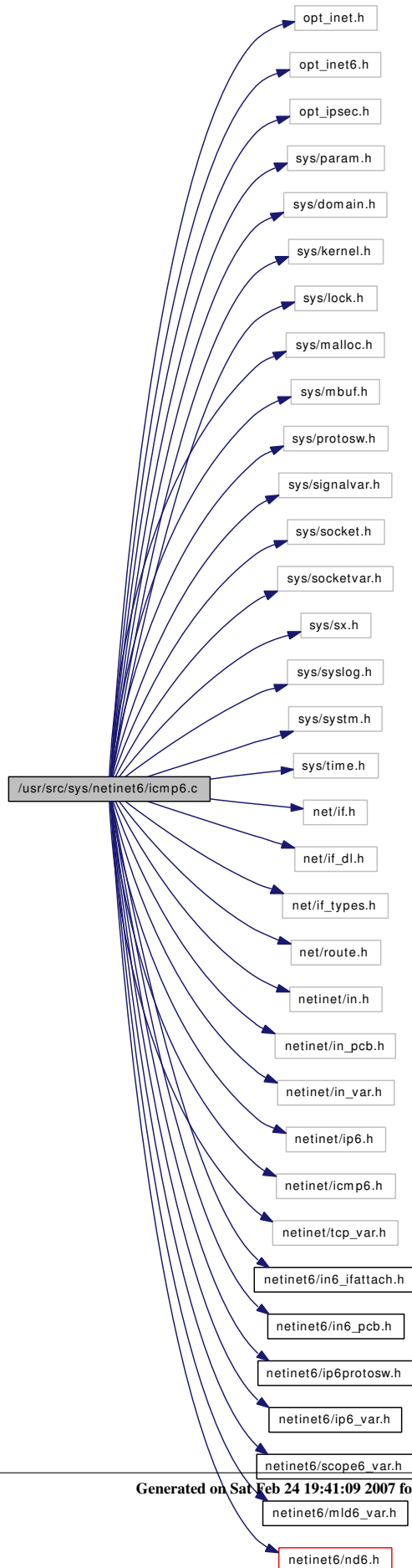
7.19.3.4 `struct mtx ip6qlock` `[static]`

Definition at line 69 of file frag6.c.

7.20 /usr/src/sys/netinet6/icmp6.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/domain.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/protosw.h>
#include <sys/signalvar.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sx.h>
#include <sys/syslog.h>
#include <sys/system.h>
#include <sys/time.h>
#include <net/if.h>
#include <net/if_dl.h>
#include <net/if_types.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_pcb.h>
#include <netinet/in_var.h>
#include <netinet/ip6.h>
#include <netinet/icmp6.h>
#include <netinet/tcp_var.h>
#include <netinet6/in6_ifattach.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/ip6protosw.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet6/mld6_var.h>
#include <netinet6/nd6.h>
```

Include dependency graph for icmp6.c:



Defines

- #define `hostnamelen` `strlen(hostname)`
- #define `hostnamelen` `strlen(hostname)`

Functions

- static void `icmp6_errcount` `__P` ((struct `icmp6errstat` *, int, int))
- static int `icmp6_rip6_input` `__P` ((struct `mbuf` **, int))
- static int `icmp6_ratelimit` `__P` ((const struct `in6_addr` *, const int, const int))
- static const char *`icmp6_redirect_diag` `__P` ((struct `in6_addr` *, struct `in6_addr` *, struct `in6_addr` **))
- static struct `mbuf` *`ni6_input` `__P` ((struct `mbuf` *, int))
- static struct `mbuf` *`ni6_nametodns` `__P` ((const char *, int, int))
- static int `ni6_dnsmatch` `__P` ((const char *, int, const char *, int))
- static int `ni6_addrs` `__P` ((struct `icmp6_nodeinfo` *, struct `mbuf` *, struct `ifnet` **, struct `in6_addr` **))
- static int `ni6_store_addrs` `__P` ((struct `icmp6_nodeinfo` *, struct `icmp6_nodeinfo` *, struct `ifnet` *, int))
- static int `icmp6_notify_error` `__P` ((struct `mbuf` **, int, int, int))
- void `icmp6_init` ()
- static void `icmp6_errcount` (struct `icmp6errstat` *`stat`, int `type`, int `code`)
- void `icmp6_error2` (struct `mbuf` *`m`, int `type`, int `code`, int `param`, struct `ifnet` *`ifp`)
- void `icmp6_error` (struct `mbuf` *`m`, int `type`, int `code`, int `param`)
- int `icmp6_input` (struct `mbuf` **`mp`, int *`offp`, int `proto`)
- static int `icmp6_notify_error` (struct `mbuf` **`mp`, int `off`, int `icmp6len`, int `code`)
- void `icmp6_mtudisc_update` (struct `ip6ctlparam` *`ip6cp`, int `validated`)
- static struct `mbuf` * `ni6_input` (struct `mbuf` *`m`, int `off`)
- static struct `mbuf` * `ni6_nametodns` (char *`name`, int `namelen`, int `old`) const
- static int `ni6_dnsmatch` (char *`a`, int `alen`, const char *`b`, int `blen`) const
- static int `ni6_addrs` (struct `icmp6_nodeinfo` *`ni6`, struct `mbuf` *`m`, struct `ifnet` **`ifpp`, struct `in6_addr` *`subj`)
- static int `ni6_store_addrs` (struct `icmp6_nodeinfo` *`ni6`, struct `icmp6_nodeinfo` *`nni6`, struct `ifnet` *`ifp0`, int `resid`)
- static int `icmp6_rip6_input` (struct `mbuf` **`mp`, int `off`)
- void `icmp6_reflect` (struct `mbuf` *`m`, `size_t` `off`)
- void `icmp6_fasttimo` ()
- static const char * `icmp6_redirect_diag` (struct `in6_addr` *`src6`, struct `in6_addr` *`dst6`, struct `in6_addr` *`tgt6`)
- void `icmp6_redirect_input` (struct `mbuf` *`m`, int `off`)
- void `icmp6_redirect_output` (struct `mbuf` *`m0`, struct `rentry` *`rt`)
- int `icmp6_ctloutput` (struct `socket` *`so`, struct `sockopt` *`sopt`)
- static int `icmp6_ratelimit` (struct `in6_addr` *`dst`, const int `type`, const int `code`) const

Variables

- domain `inet6domain`
- `icmp6stat` `icmp6stat`
- `inpcbinfo` `ripcbinfo`
- `inpcbhead` `ripcb`
- int `icmp6errppslim`
- static int `icmp6errpps_count` = 0
- static struct `timeval` `icmp6errppslim_last`
- int `icmp6_nodeinfo`

7.20.1 Define Documentation

7.20.1.1 #define hostnamelen strlen(hostname)

Definition at line 1155 of file icmp6.c.

7.20.1.2 #define hostnamelen strlen(hostname)

Definition at line 1155 of file icmp6.c.

Referenced by get_rand_ifid(), icmp6_input(), in6_update_ifa(), and ni6_input().

7.20.2 Function Documentation

7.20.2.1 static int icmp6_notify_error __P ((struct mbuf **, int, int, int)) [static]

7.20.2.2 static int ni6_store_addr __P ((struct icmp6_nodeinfo *, struct icmp6_nodeinfo *, struct ifnet *, int)) [static]

7.20.2.3 static int ni6_addr __P ((struct icmp6_nodeinfo *, struct mbuf *, struct ifnet **, struct in6_addr *)) [static]

7.20.2.4 static int ni6_dnsmatch __P ((const char *, int, const char *, int)) [static]

7.20.2.5 static struct mbuf* ni6_nametodns __P ((const char *, int, int)) [static]

7.20.2.6 static struct mbuf* ni6_input __P ((struct mbuf *, int)) [static]

7.20.2.7 static const char* icmp6_redirect_diag __P ((struct in6_addr *, struct in6_addr *, struct in6_addr *)) [static]

7.20.2.8 static int icmp6_ratelimit __P ((const struct in6_addr *, const int, const int)) [static]

7.20.2.9 static int icmp6_rip6_input __P ((struct mbuf **, int)) [static]

7.20.2.10 static void icmp6_errcount __P ((struct icmp6errstat *, int, int)) [static]

7.20.2.11 int icmp6_ctloutput (struct socket * so, struct sockopt * sopt)

Definition at line 2699 of file icmp6.c.

References ICMP6_FILTER.

Referenced by rip6_ctloutput().

7.20.2.12 static void icmp6_errcount (struct icmp6errstat * stat, int type, int code) [static]

Definition at line 145 of file icmp6.c.

Referenced by icmp6_error(), and icmp6_redirect_output().

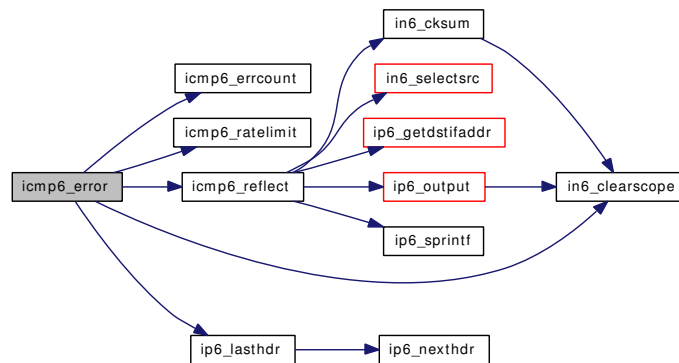
7.20.2.13 void icmp6_error (struct mbuf * m, int type, int code, int param)

Definition at line 241 of file icmp6.c.

References icmp6_errcount(), icmp6_ratelimit(), icmp6_reflect(), icmp6stat, in6_clearscope(), IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, ip6_lasthdr(), M_DECRYPTED, and nd6log.

Referenced by frag6_freef(), frag6_input(), icmp6_error2(), ip6_forward(), ip6_input(), ip6_process_hopopts(), ip6_rthdr0(), ip6_unknown_opt(), phyint_send(), rip6_input(), route6_input(), and udp6_input().

Here is the call graph for this function:



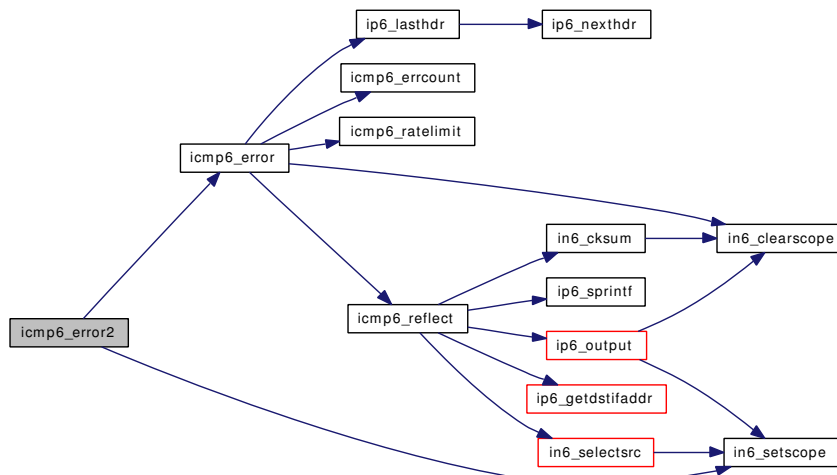
7.20.2.14 void icmp6_error2 (struct mbuf * m, int type, int code, int param, struct ifnet * ifp)

Definition at line 207 of file icmp6.c.

References icmp6_error(), and in6_setscope().

Referenced by nd6_llinfo_timer().

Here is the call graph for this function:



7.20.2.15 void icmp6_fasttimo ()

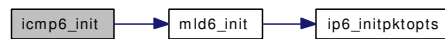
Definition at line 2186 of file icmp6.c.

7.20.2.16 void icmp6_init ()

Definition at line 139 of file icmp6.c.

References mld6_init().

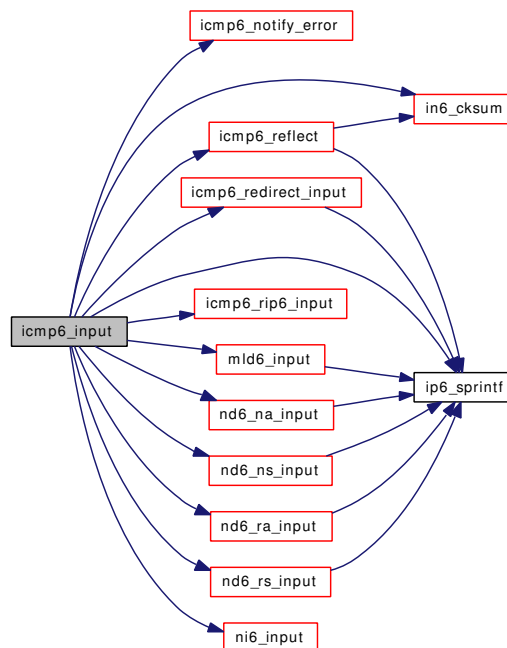
Here is the call graph for this function:

**7.20.2.17 int icmp6_input (struct mbuf ** mp, int * offp, int proto)**

Definition at line 397 of file icmp6.c.

References faithprefix_p, hostnamelen, icmp6_nodeinfo, icmp6_notify_error(), icmp6_redirect_input(), icmp6_reflect(), icmp6_rip6_input(), icmp6stat, in6_cksum(), INET6_ADDRSTRLEN, ip6_sprintf(), mld6_input(), nd6_na_input(), nd6_ns_input(), nd6_ra_input(), nd6_rs_input(), nd6log, and ni6_input().

Here is the call graph for this function:

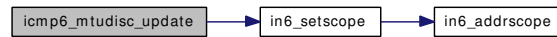
**7.20.2.18 void icmp6_mtudisc_update (struct ip6ctlparam * ip6cp, int validated)**

Definition at line 1099 of file icmp6.c.

References icmp6stat, and in6_setscope().

Referenced by icmp6_notify_error().

Here is the call graph for this function:



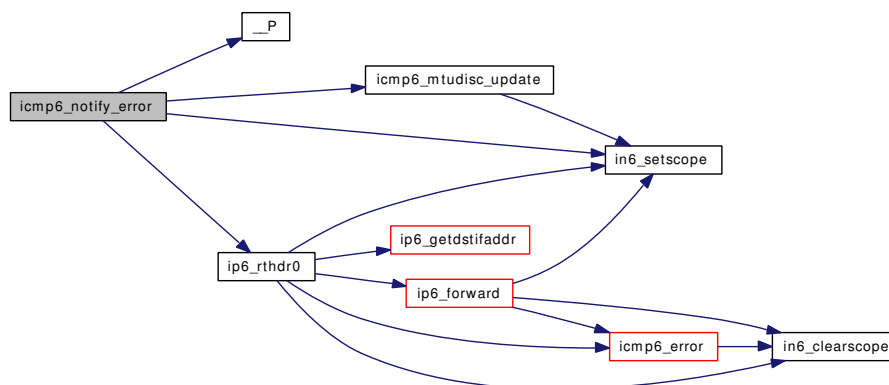
7.20.2.19 static int icmp6_notify_error (struct mbuf ** mp, int off, int icmp6len, int code) [static]

Definition at line 866 of file icmp6.c.

References __P(), icmp6_mtudisc_update(), icmp6stat, in6_setscope(), inet6sw, ip6_protox, ip6_rthdr0(), ip6ctlparam::ip6c_cmdarg, ip6ctlparam::ip6c_finaldst, ip6ctlparam::ip6c_icmp6, ip6ctlparam::ip6c_ip6, ip6ctlparam::ip6c_m, ip6ctlparam::ip6c_nxt, ip6ctlparam::ip6c_off, ip6ctlparam::ip6c_src, and IPV6_RTHDR_TYPE_0.

Referenced by icmp6_input().

Here is the call graph for this function:



7.20.2.20 static int icmp6_ratelimit (struct in6_addr * dst, const int type, const int code) const [static]

Definition at line 2778 of file icmp6.c.

References icmp6errpps_count, icmp6errppslim, and icmp6errppslim_last.

Referenced by icmp6_error(), and icmp6_redirect_output().

7.20.2.21 static const char* icmp6_redirect_diag (struct in6_addr * src6, struct in6_addr * dst6, struct in6_addr * tgt6) [static]

Definition at line 2193 of file icmp6.c.

References INET6_ADDRSTRLEN, and ip6_sprintf().

Referenced by icmp6_redirect_input().

Here is the call graph for this function:



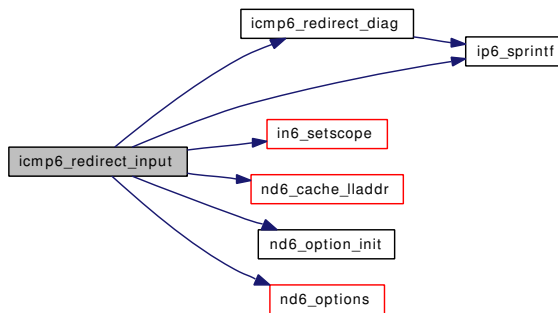
7.20.2.22 void icmp6_redirect_input (struct mbuf * m, int off)

Definition at line 2209 of file icmp6.c.

References icmp6_rediraccept, icmp6_redirect_diag(), icmp6stat, IN6_IS_ADDR_LINKLOCAL, IN6_IS_ADDR_MULTICAST, in6_setscope(), INET6_ADDRSTRLEN, ip6_forwarding, ip6_sprintf(), nd6_cache_lladdr(), nd6_option_init(), nd6_options(), nd6log, and sin6.

Referenced by icmp6_input().

Here is the call graph for this function:



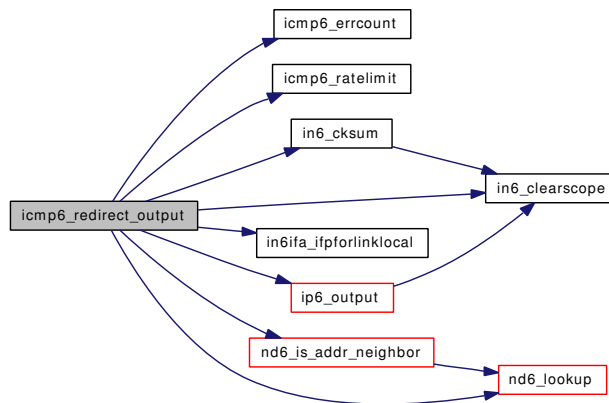
7.20.2.23 void icmp6_redirect_output (struct mbuf * m0, struct rentry * rt)

Definition at line 2413 of file icmp6.c.

References in6_ifaddr::ia_addr, icmp6_errcount(), icmp6_ratelimit(), icmp6stat, in6_cksum(), in6_clearscope(), IN6_IFF_ANYCAST, IN6_IFF_NOTREADY, IN6_IS_ADDR_LINKLOCAL, IN6_IS_ADDR_MULTICAST, in6ifa_ifpforlinklocal(), ip6_forwarding, ip6_output(), M_DECRYPTED, nd6_is_addr_neighbor(), nd6_lookup(), sin6, and sockaddr_in6::sin6_addr.

Referenced by ip6_forward().

Here is the call graph for this function:



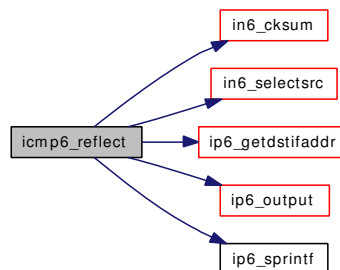
7.20.2.24 void icmp6_reflect (struct mbuf * m, size_t off)

Definition at line 2026 of file icmp6.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia_addr, in6_cksum(), IN6_IFF_ANYCAST, IN6_IFF_NOTREADY, IN6_IS_ADDR_MULTICAST, in6_selectsrc(), INET6_ADDRSTRLEN, ip6_defhlim, ip6_getdstifaddr(), ip6_output(), ip6_sprintf(), nd6log, ND_IFINFO, sin6, and sockaddr_in6::sin6_addr.

Referenced by icmp6_error(), and icmp6_input().

Here is the call graph for this function:



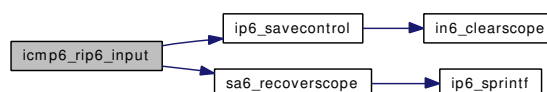
7.20.2.25 static int icmp6_rip6_input (struct mbuf ** mp, int off) [static]

Definition at line 1862 of file icmp6.c.

References IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_UNSPECIFIED, in6pcb, ip6_savecontrol(), ripcb, ripcbinfo, and sa6_recoverscope().

Referenced by icmp6_input().

Here is the call graph for this function:



7.20.2.26 `static int ni6_addr (struct icmp6_nodeinfo * ni6, struct mbuf * m, struct ifnet ** ifpp, struct in6_addr * subj) [static]`

Definition at line 1626 of file icmp6.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia_addr`, `icmp6_nodeinfo`, `in6_addrscope()`, `IN6_ARE_ADDR_EQUAL`, `IN6_IFF_ANYCAST`, `IN6_IFF_TEMPORARY`, `IPV6_ADDR_SCOPE_GLOBAL`, `IPV6_ADDR_SCOPE_LINKLOCAL`, `IPV6_ADDR_SCOPE_SITELOCAL`, and `sockaddr_in6::sin6_addr`.

Referenced by `ni6_input()`.

Here is the call graph for this function:



7.20.2.27 `static int ni6_dnsmatch (char * a, int alen, const char * b, int blen) const [static]`

Definition at line 1562 of file icmp6.c.

Referenced by `ni6_input()`.

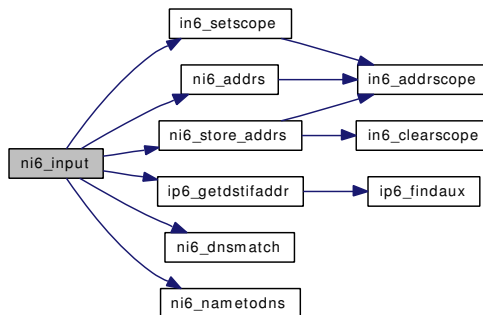
7.20.2.28 `static struct mbuf* ni6_input (struct mbuf * m, int off) [static]`

Definition at line 1157 of file icmp6.c.

References `hostnamelen`, `in6_ifaddr::ia6_flags`, `icmp6_nodeinfo`, `IN6_ARE_ADDR_EQUAL`, `IN6_IFF_TEMPORARY`, `IN6_IS_ADDR_MC_LINKLOCAL`, `IN6_IS_ADDR_MULTICAST`, `in6_setscope()`, `ip6_getdstifaddr()`, `nd6log`, `ni6_addr()`, `ni6_dnsmatch()`, `ni6_nametodns()`, and `ni6_store_addr()`.

Referenced by `icmp6_input()`.

Here is the call graph for this function:



7.20.2.29 `static struct mbuf* ni6_nametodns (char * name, int namelen, int old) const [static]`

Definition at line 1462 of file icmp6.c.

Referenced by `ni6_input()`.

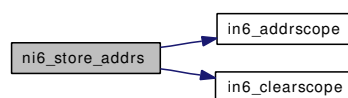
7.20.2.30 `static int ni6_store_addrs (struct icmp6_nodeinfo * ni6, struct icmp6_nodeinfo * nni6, struct ifnet * ifp0, int resid)` [static]

Definition at line 1720 of file icmp6.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia6_lifetime`, `in6_addrlifetime::ia6t_expire`, `in6_ifaddr::ia_addr`, `icmp6_nodeinfo`, `in6_addrscope()`, `in6_clearscope()`, `IN6_IFF_ANYCAST`, `IN6_IFF_DEPRECATED`, `IN6_IFF_TEMPORARY`, `IPV6_ADDR_SCOPE_GLOBAL`, `IPV6_ADDR_SCOPE_LINKLOCAL`, `IPV6_ADDR_SCOPE_SITELOCAL`, `ND6_INFINITE_LIFETIME`, and `sockaddr_in6::sin6_addr`.

Referenced by `ni6_input()`.

Here is the call graph for this function:



7.20.3 Variable Documentation

7.20.3.1 int icmp6_nodeinfo

Definition at line 446 of file in6_proto.c.

Referenced by `icmp6_input()`, `ni6_addrs()`, `ni6_input()`, and `ni6_store_addrs()`.

7.20.3.2 int icmp6errpps_count = 0

 [static]

Definition at line 119 of file icmp6.c.

Referenced by `icmp6_ratelimit()`.

7.20.3.3 int icmp6errppslim

Definition at line 445 of file in6_proto.c.

Referenced by `icmp6_ratelimit()`.

7.20.3.4 struct timeval icmp6errppslim_last

 [static]

Definition at line 120 of file icmp6.c.

Referenced by `icmp6_ratelimit()`.

7.20.3.5 struct icmp6stat icmp6stat

Definition at line 114 of file icmp6.c.

Referenced by `icmp6_error()`, `icmp6_input()`, `icmp6_mtudisc_update()`, `icmp6_notify_error()`, `icmp6_redirect_input()`, `icmp6_redirect_output()`, `mld6_input()`, `mld6_sendpkt()`, `nd6_na_input()`, `nd6_na_output()`, `nd6_ns_input()`, `nd6_ns_output()`, `nd6_options()`, `nd6_ra_input()`, `nd6_rs_input()`, and `rip6_output()`.

7.20.3.6 struct domain [inet6domain](#)

Definition at line 369 of file in6_proto.c.

Referenced by ip6_init().

7.20.3.7 struct inpcbhead [ripcb](#)

Referenced by icmp6_rip6_input(), and rip6_input().

7.20.3.8 struct inpcbinfo [ripcbinfo](#)

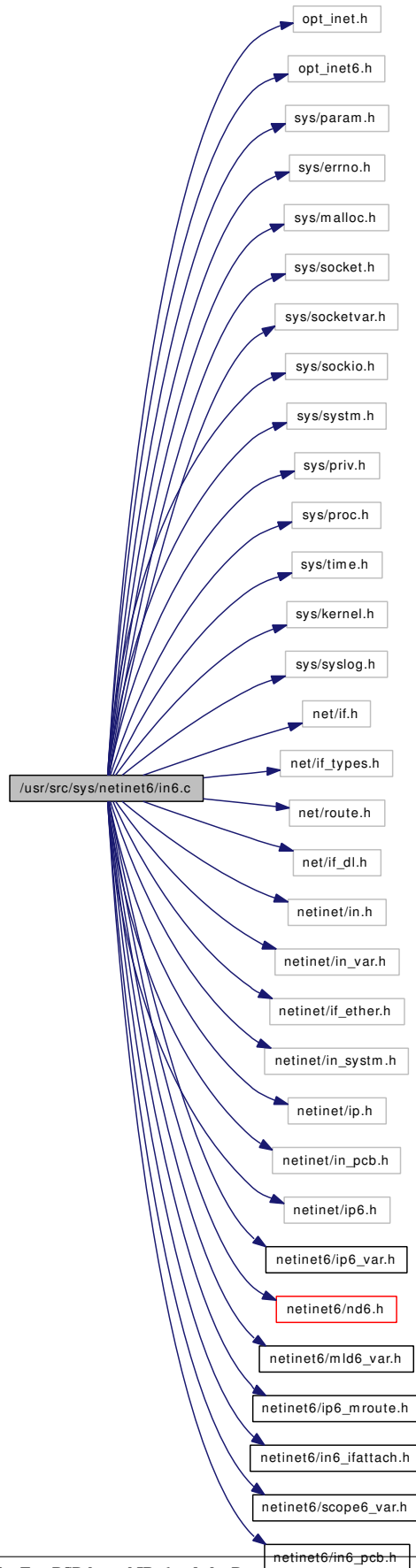
Referenced by icmp6_rip6_input(), in6_ifdetach(), rip6_attach(), rip6_bind(), rip6_connect(), rip6_ctlinput(), rip6_detach(), rip6_input(), and rip6_send().

7.21 /usr/src/sys/netinet6/icmp6.h File Reference

7.22 /usr/src/sys/netinet6/in6.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/errno.h>
#include <sys/malloc.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sockio.h>
#include <sys/system.h>
#include <sys/priv.h>
#include <sys/proc.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/route.h>
#include <net/if_dl.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/if_ether.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/in_pcb.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/nd6.h>
#include <netinet6/mld6_var.h>
#include <netinet6/ip6_mroute.h>
#include <netinet6/in6_ifattach.h>
#include <netinet6/scope6_var.h>
#include <netinet6/in6_pcb.h>
```

Include dependency graph for in6.c:



Defines

- #define `ifa2ia6(ifa)` ((struct `in6_ifaddr *`)(ifa))
- #define `ia62ifa(ia6)` (&((ia6) → ia_ifa))
- #define `MLTMASK_LEN` 4
- #define `hostnamelen` strlen(hostname)

Functions

- `MALLOC_DEFINE` (M_IP6MADDR, "in6_multi", "internet multicast address")
- static int `in6_lifaddr_ioctl` `__P` ((struct socket *, u_long, caddr_t, struct ifnet *, struct thread *)
- static int `in6_ifinit` `__P` ((struct ifnet *, struct `in6_ifaddr *`, struct `sockaddr_in6 *`, int))
- static void `in6_unlink_ifa` `__P` ((struct `in6_ifaddr *`, struct ifnet *)
- static void `in6_ifloop_request` (int cmd, struct ifaddr *ifa)
- void `in6_ifaddloop` (struct ifaddr *ifa)
- void `in6_ifremloop` (struct ifaddr *ifa)
- int `in6_mask2len` (struct `in6_addr *`mask, u_char *lim0)
- int `in6_control` (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- int `in6_update_ifa` (struct ifnet *ifp, struct `in6_aliasreq *`ifra, struct `in6_ifaddr *`ia, int flags)
- void `in6_purgeaddr` (struct ifaddr *ifa)
- static void `in6_unlink_ifa` (struct `in6_ifaddr *`ia, struct ifnet *ifp)
- void `in6_purgeif` (struct ifnet *ifp)
- static int `in6_lifaddr_ioctl` (struct socket *so, u_long cmd, caddr_t data, struct ifnet *ifp, struct thread *td)
- static int `in6_ifinit` (struct ifnet *ifp, struct `in6_ifaddr *`ia, struct `sockaddr_in6 *`sin6, int newhost)
- `in6_multi_mship *` `in6_joiningroup` (struct ifnet *ifp, struct `in6_addr *`addr, int *errorp, int delay)
- int `in6_leavegroup` (struct `in6_multi_mship *`imm)
- `in6_ifaddr *` `in6ifa_ifpforlinklocal` (struct ifnet *ifp, int ignoreflags)
- `in6_ifaddr *` `in6ifa_ifpwithaddr` (struct ifnet *ifp, struct `in6_addr *`addr)
- char * `ip6_sprintf` (char *ip6buf, const struct `in6_addr *`addr)
- int `in6_localaddr` (struct `in6_addr *`in6)
- int `in6_is_addr_deprecated` (struct `sockaddr_in6 *`sa6)
- int `in6_matchlen` (struct `in6_addr *`src, struct `in6_addr *`dst)
- int `in6_are_prefix_equal` (struct `in6_addr *`p1, struct `in6_addr *`p2, int len)
- void `in6_prefixlen2mask` (struct `in6_addr *`maskp, int len)
- `in6_ifaddr *` `in6_ifawithifp` (struct ifnet *ifp, struct `in6_addr *`dst)
- void `in6_if_up` (struct ifnet *ifp)
- int `in6if_do_dad` (struct ifnet *ifp)
- void `in6_setmaxmtu` ()
- int `in6_if2idlen` (struct ifnet *ifp)
- void * `in6_domifattach` (struct ifnet *ifp)
- void `in6_domifdetach` (struct ifnet *ifp, void *aux)
- void `in6_sin6_2_sin` (struct `sockaddr_in *`sin, struct `sockaddr_in6 *`sin6)
- void `in6_sin_2_v4mapsin6` (struct `sockaddr_in *`sin, struct `sockaddr_in6 *`sin6)
- void `in6_sin6_2_sin_in_sock` (struct `sockaddr *`nam)
- void `in6_sin_2_v4mapsin6_in_sock` (struct `sockaddr **`nam)

Variables

- `in6_addr in6addr_any` = IN6ADDR_ANY_INIT
- `in6_addr in6addr_loopback` = IN6ADDR_LOOPBACK_INIT
- `in6_addr in6addr_nodelocal_allnodes`
- `in6_addr in6addr_linklocal_allnodes`
- `in6_addr in6addr_linklocal_allrouters`
- `in6_addr in6mask0` = IN6MASK0
- `in6_addr in6mask32` = IN6MASK32
- `in6_addr in6mask64` = IN6MASK64
- `in6_addr in6mask96` = IN6MASK96
- `in6_addr in6mask128` = IN6MASK128
- `sockaddr_in6 sa6_any`
- `in6_multihead in6_multihead`
- `int(*) faithprefix_p` (struct `in6_addr *`)
- static char `digits []` = "0123456789abcdef"

7.22.1 Define Documentation

7.22.1.1 `#define hostnamelen strlen(hostname)`

7.22.1.2 `#define ia62ifa(ia6) (&((ia6) → ia_ifa))`

Definition at line 318 of file `in6.c`.

7.22.1.3 `#define ifa2ia6(ifa) ((struct in6_ifaddr *) (ifa))`

Definition at line 317 of file `in6.c`.

Referenced by `in6_lifaddr_ioctl()`.

7.22.1.4 `#define MLTMASK_LEN 4`

Referenced by `in6_update_ifa()`.

7.22.2 Function Documentation

7.22.2.1 `static void in6_unlink_ifa __P ((struct in6_ifaddr *, struct ifnet *))` [static]

7.22.2.2 `static int in6_ifinit __P ((struct ifnet *, struct in6_ifaddr *, struct sockaddr_in6 *, int))`
[static]

7.22.2.3 `static int in6_lifaddr_ioctl __P ((struct socket *, u_long, caddr_t, struct ifnet *, struct thread *))` [static]

7.22.2.4 `int in6_are_prefix_equal (struct in6_addr * p1, struct in6_addr * p2, int len)`

Definition at line 2023 of file `in6.c`.

Referenced by `nd6_prefix_lookup()`, `nd6_prefix_offlink()`, and `nd6_prefix_onlink()`.

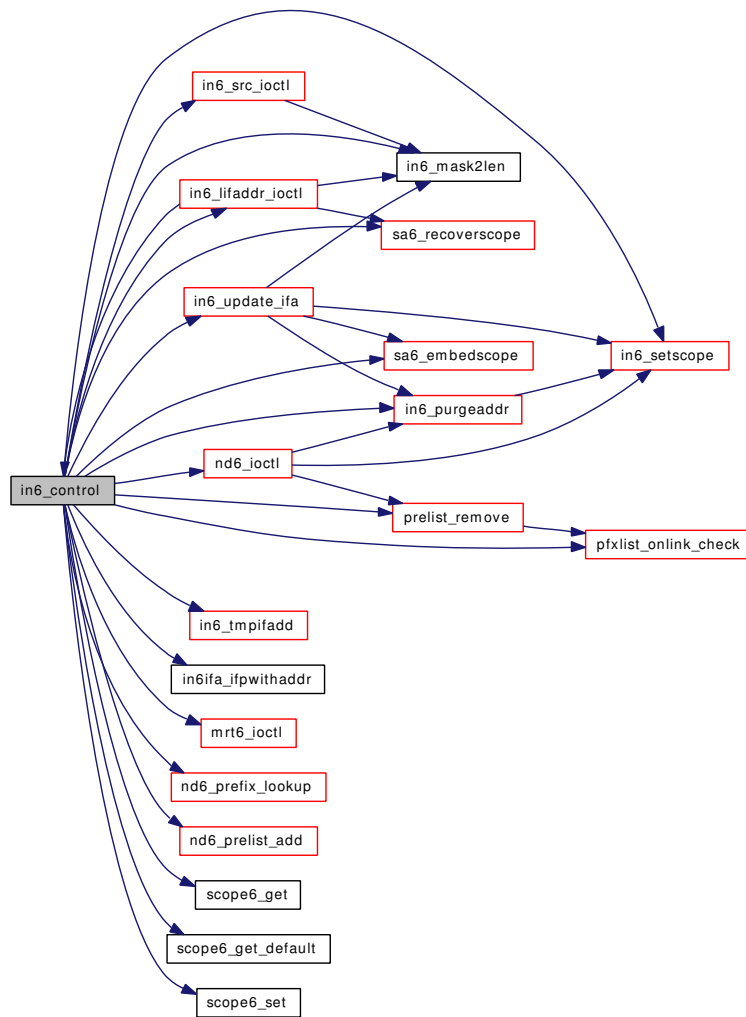
7.22.2.5 `int in6_control (struct socket * so, u_long cmd, caddr_t data, struct ifnet * ifp, struct thread * td)`

Definition at line 321 of file in6.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia6_lifetime`, `in6_ifaddr::ia6_ndpr`, `in6_ifaddr::ia6_update_time`, `in6_addrlifetime::ia6t_expire`, `in6_addrlifetime::ia6t_pltime`, `in6_addrlifetime::ia6t_preferred`, `in6_addrlifetime::ia6t_vltime`, `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_dstaddr`, `in6_ifaddr::ia_ifa`, `in6_ifaddr::ia_prefixmask`, `in6_ifreq::ifr_ifru`, `in6_aliasreq::ifra_addr`, `in6_aliasreq::ifra_flags`, `in6_aliasreq::ifra_lifetime`, `in6_aliasreq::ifra_prefixmask`, `in6_ifreq::ifru_flags6`, `in6_ifreq::ifru_icmp6stat`, `in6_ifreq::ifru_lifetime`, `in6_ifreq::ifru_scope_id`, `in6_ifreq::ifru_stat`, `IN6_IFF_AUTOCONF`, `in6_lifaddr_ioctl()`, `in6_mask2len()`, `in6_purgeaddr()`, `in6_setscope()`, `in6_src_ioctl()`, `in6_tmpifadd()`, `in6_update_ifa()`, `in6ifa_ifpwithaddr()`, `ip6_use_tempaddr`, `mrt6_ioctl()`, `ND6_INFINITE_LIFETIME`, `nd6_ioctl()`, `nd6_prefix_lookup()`, `nd6_prelist_add()`, `nd_prefix::ndpr_refcnt`, `OSIOCGIFINFO_IN6`, `pfxlist_onlink_check()`, `prelist_remove()`, `sa6_embedscope()`, `sa6_recoverscope()`, `scope6_get()`, `scope6_get_default()`, `scope6_set()`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, `sockaddr_in6::sin6_len`, `sockaddr_in6::sin6_scope_id`, `SIOCAADDRCTL_POLICY`, `SIOCAIFADDR_IN6`, `SIOCAIFPREFIX_IN6`, `SIOCCIFPREFIX_IN6`, `SIOCADDRCTL_POLICY`, `SIOCDEFADDR_IN6`, `SIOCDEFPREFIX_IN6`, `SIOCGDEFIFACE_IN6`, `SIOCGDRLST_IN6`, `SIOCGETMIFCNT_IN6`, `SIOCGSGCNT_IN6`, `SIOCGIFADDR_IN6`, `SIOCGIFAFLAG_IN6`, `SIOCGIFALIFETIME_IN6`, `SIOCGIFDSTADDR_IN6`, `SIOCGIFINFO_IN6`, `SIOCGIFNETMASK_IN6`, `SIOCGIFPDSTADDR_IN6`, `SIOCGIFPREFIX_IN6`, `SIOCGIFPSRCADDR_IN6`, `SIOCGIFSTAT_ICMP6`, `SIOCGIFSTAT_IN6`, `SIOCGNBRINFO_IN6`, `SIOCGPRLST_IN6`, `SIOCGSCOPE6`, `SIOCGSCOPE6DEF`, `SIOCSDEFIFACE_IN6`, `SIOCSGIFPREFIX_IN6`, `SIOCSIFADDR_IN6`, `SIOCSIFALIFETIME_IN6`, `SIOCSIFDSTADDR_IN6`, `SIOCSIFINFO_FLAGS`, `SIOCSIFINFO_IN6`, `SIOCSIFNETMASK_IN6`, `SIOCSIFPHYADDR_IN6`, `SIOCSIFPREFIX_IN6`, `SIOCSNDFLUSH_IN6`, `SIOCSFPXFLUSH_IN6`, `SIOCSRTRFLUSH_IN6`, and `SIOCSCOPE6`.

Referenced by `in6_lifaddr_ioctl()`.

Here is the call graph for this function:

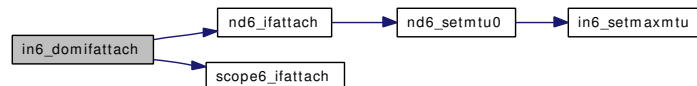


7.22.2.6 void* in6_domifattach (struct ifnet * ifp)

Definition at line 2310 of file in6.c.

References `nd6_ifattach()`, and `scope6_ifattach()`.

Here is the call graph for this function:

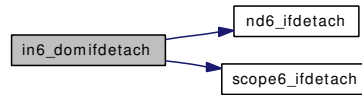


7.22.2.7 void in6_domifdetach (struct ifnet * ifp, void * aux)

Definition at line 2333 of file in6.c.

References `in6_ifextra::icmp6_ifstat`, `in6_ifextra::in6_ifstat`, `nd6_ifdetach()`, `in6_ifextra::nd_ifinfo`, `in6_ifextra::scope6_id`, and `scope6_ifdetach()`.

Here is the call graph for this function:



7.22.2.8 `int in6_if2idlen (struct ifnet * ifp)`

Definition at line 2258 of file `in6.c`.

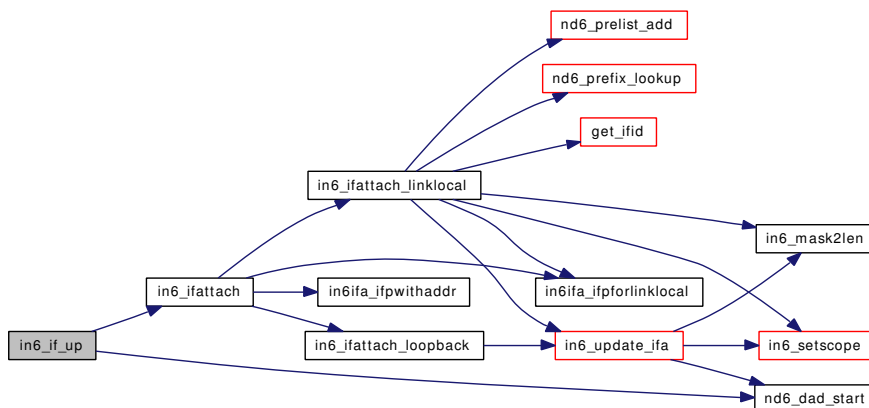
Referenced by `prelist_update()`.

7.22.2.9 `void in6_if_up (struct ifnet * ifp)`

Definition at line 2160 of file `in6.c`.

References `in6_ifaddr::ia6_flags`, `in6_ifattach()`, `IN6_IFF_TENTATIVE`, `MAX_RTR_SOLICITATION_DELAY`, and `nd6_dad_start()`.

Here is the call graph for this function:



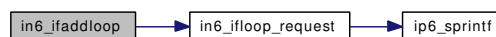
7.22.2.10 `void in6_ifaddloop (struct ifaddr * ifa)`

Definition at line 209 of file `in6.c`.

References `in6_ifloop_request()`.

Referenced by `in6_ifinit()`.

Here is the call graph for this function:



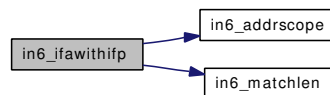
7.22.2.11 struct `in6_ifaddr`* `in6_ifawithifp` (struct `ifnet` * `ifp`, struct `in6_addr` * `dst`)

Definition at line 2078 of file `in6.c`.

References `in6_ifaddr::ia6_flags`, `IFA_IN6`, `in6_addrscope()`, `IN6_IFF_ANYCAST`, `IN6_IFF_DEPRECATED`, `IN6_IFF_DETACHED`, `IN6_IFF_NOTREADY`, `in6_matchlen()`, and `ip6_use_deprecated`.

Referenced by `ip6_input()`, and `ip6_output()`.

Here is the call graph for this function:



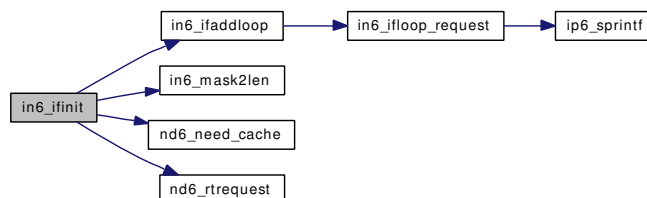
7.22.2.12 static int `in6_ifinit` (struct `ifnet` * `ifp`, struct `in6_ifaddr` * `ia`, struct `sockaddr_in6` * `sin6`, int `newhost`) [static]

Definition at line 1709 of file `in6.c`.

References `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_dstaddr`, `in6_ifaddr::ia_ifa`, `in6_ifaddr::ia_prefixmask`, `in6_ifaddloop()`, `in6_mask2len()`, `llinfo_nd6::ln_state`, `ND6_LLINFO_STALE`, `nd6_need_cache()`, `nd6_rtrequest()`, `sin6`, `sockaddr_in6::sin6_addr`, and `sockaddr_in6::sin6_family`.

Referenced by `in6_update_ifa()`.

Here is the call graph for this function:



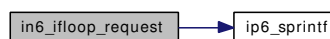
7.22.2.13 static void `in6_ifloop_request` (int `cmd`, struct `ifaddr` * `ifa`) [static]

Definition at line 138 of file `in6.c`.

References `all1_sa`, `in6mask128`, `INET6_ADDRSTRLEN`, and `ip6_sprintf()`.

Referenced by `in6_ifaddloop()`.

Here is the call graph for this function:



7.22.2.14 void in6_ifremloop (struct ifaddr * ifa)

Definition at line 229 of file in6.c.

References in6_ifaddr::ia_addr, in6_ifaddr::ia_next, IFA_IN6, IN6_ARE_ADDR_EQUAL, and sockaddr_in6::sin6_addr.

Referenced by in6_purgeaddr().

7.22.2.15 int in6_is_addr_deprecated (struct sockaddr_in6 * sa6)

Definition at line 1980 of file in6.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia_addr, in6_ifaddr::ia_next, IN6_ARE_ADDR_EQUAL, IN6_IFF_DEPRECATED, and sockaddr_in6::sin6_addr.

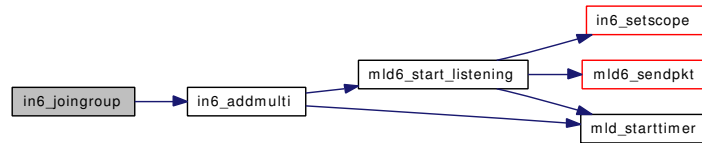
7.22.2.16 struct in6_multi_mship* in6_joingroup (struct ifnet * ifp, struct in6_addr * addr, int * errorp, int delay)

Definition at line 1817 of file in6.c.

References in6_addmulti().

Referenced by in6_update_ifa(), and ip6_setmoptions().

Here is the call graph for this function:

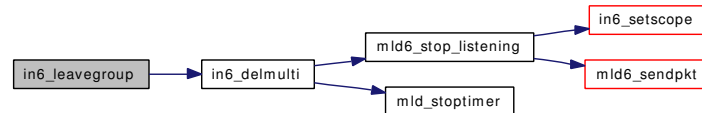


7.22.2.17 int in6_leavegroup (struct in6_multi_mship * imm)

Definition at line 1840 of file in6.c.

References in6_delmulti().

Here is the call graph for this function:



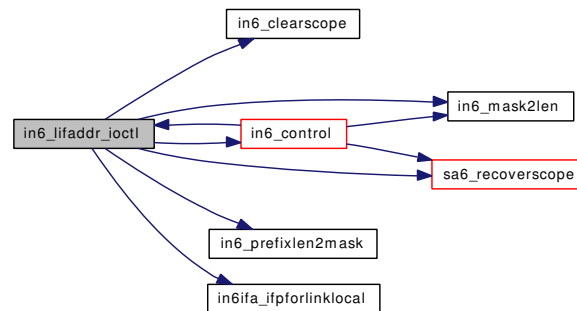
7.22.2.18 static int in6_lifaddr_ioctl (struct socket * so, u_long cmd, caddr_t data, struct ifnet * ifp, struct thread * td) [static]

Definition at line 1471 of file in6.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_dstaddr`, `in6_ifaddr::ia_prefixmask`, `ifa2ia6`, `IFA_IN6`, `IN6_ARE_ADDR_EQUAL`, `in6_clearscope()`, `in6_control()`, `in6_mask2len()`, `in6_prefixlen2mask()`, `in6ifa_ifpforlinklocal()`, `sa6_recoverscope()`, `sin6`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_len`, `SIOCAIFADDR_IN6`, and `SIOCIFADDR_IN6`.

Referenced by `in6_control()`.

Here is the call graph for this function:



7.22.2.19 int in6_localaddr (struct in6_addr * in6)

Definition at line 1961 of file `in6.c`.

References `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_next`, `in6_ifaddr::ia_prefixmask`, `IN6_ARE_MASKED_ADDR_EQUAL`, `IN6_IS_ADDR_LINKLOCAL`, `IN6_IS_ADDR_LOOPBACK`, and `sockaddr_in6::sin6_addr`.

Referenced by `ip6_output()`.

7.22.2.20 int in6_mask2len (struct in6_addr * mask, u_char * lim0)

Definition at line 280 of file `in6.c`.

Referenced by `in6_control()`, `in6_ifadd()`, `in6_ifattach_linklocal()`, `in6_ifinit()`, `in6_lifaddr_ioctl()`, `in6_src_ioctl()`, and `in6_update_ifa()`.

7.22.2.21 int in6_matchlen (struct in6_addr * src, struct in6_addr * dst)

Definition at line 2002 of file `in6.c`.

Referenced by `in6_ifawithifp()`.

7.22.2.22 void in6_prefixlen2mask (struct in6_addr * maskp, int len)

Definition at line 2050 of file `in6.c`.

Referenced by `in6_ifadd()`, `in6_lifaddr_ioctl()`, and `nd6_prelist_add()`.

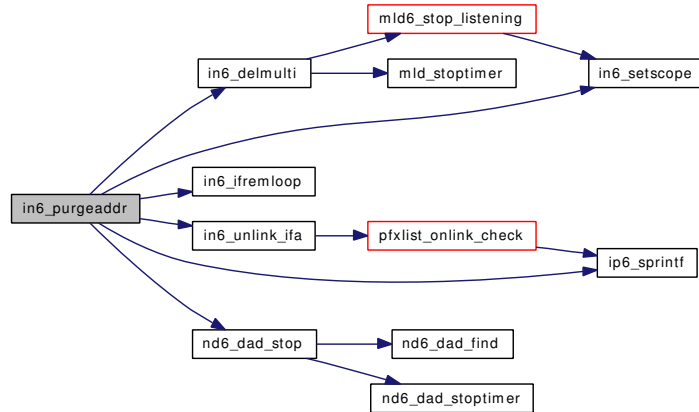
7.22.2.23 void in6_purgeaddr (struct ifaddr * ifa)

Definition at line 1321 of file `in6.c`.

References `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_dstaddr`, `in6_ifaddr::ia_ifa`, `in6_delmulti()`, `in6_ifremloop()`, `IN6_LOOKUP_MULTI`, `in6_setscope()`, `in6_unlink_ifa()`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `IPV6_ADDR_INT32_MLL`, `nd6_dad_stop()`, `sockaddr_in6::sin6_addr`, and `sockaddr_in6::sin6_len`.

Referenced by `in6_control()`, `in6_ifdetach()`, `in6_purgeif()`, `in6_update_ifa()`, `nd6_ioctl()`, and `nd6_timer()`.

Here is the call graph for this function:

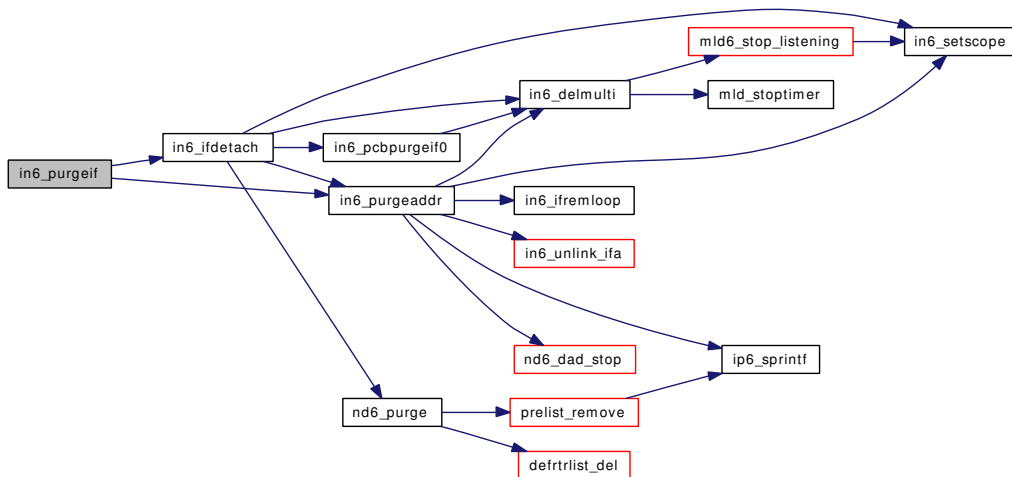


7.22.2.24 void in6_purgeif (struct ifnet * ifp)

Definition at line 1432 of file `in6.c`.

References `in6_ifdetach()`, and `in6_purgeaddr()`.

Here is the call graph for this function:



7.22.2.25 void in6_setmaxmtu ()

Definition at line 2230 of file `in6.c`.

References IN6_LINKMTU, and in6_maxmtu.

Referenced by nd6_ra_input(), and nd6_setmtu0().

7.22.2.26 void in6_sin6_2_sin (struct sockaddr_in * sin, struct sockaddr_in6 * sin6)

Definition at line 2351 of file in6.c.

References sin6, sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_port.

Referenced by in6_pcbbind(), in6_sin6_2_sin_in_sock(), sctp6_bind(), sctp6_connect(), sctp6_send(), udp6_bind(), and udp6_connect().

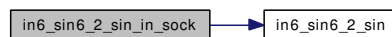
7.22.2.27 void in6_sin6_2_sin_in_sock (struct sockaddr * nam)

Definition at line 2376 of file in6.c.

References in6_sin6_2_sin(), and sin6.

Referenced by udp6_send().

Here is the call graph for this function:



7.22.2.28 void in6_sin_2_v4mapsin6 (struct sockaddr_in * sin, struct sockaddr_in6 * sin6)

Definition at line 2362 of file in6.c.

References IPV6_ADDR_INT32_SMP, and sin6.

Referenced by in6_sin_2_v4mapsin6_in_sock(), in6_v4mapsin6_sockaddr(), sctp6_getpeeraddr(), and sctp6_in6getaddr().

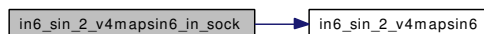
7.22.2.29 void in6_sin_2_v4mapsin6_in_sock (struct sockaddr ** nam)

Definition at line 2392 of file in6.c.

References in6_sin_2_v4mapsin6().

Referenced by in6_mapped_peeraddr(), and in6_mapped_sockaddr().

Here is the call graph for this function:



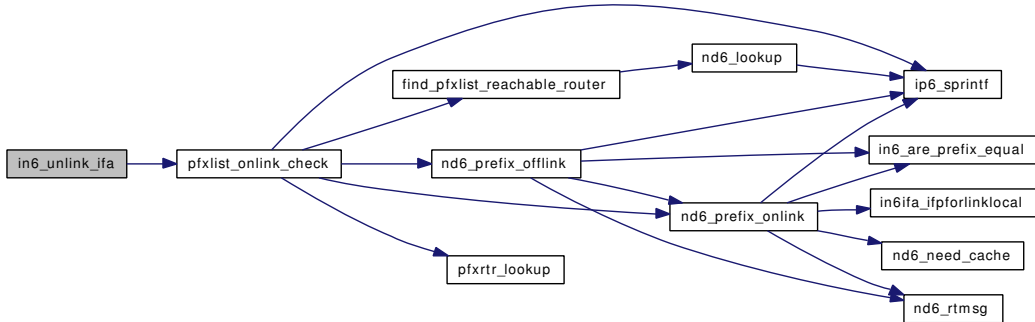
7.22.2.30 static void in6_unlink_ifa (struct in6_ifaddr * ia, struct ifnet * ifp) [static]

Definition at line 1377 of file in6.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia6_ndpr, in6_ifaddr::ia_ifa, in6_ifaddr::ia_next, IN6_IFF_AUTOCONF, nd6log, nd_prefix::ndpr_refcnt, and pfxlist_onlink_check().

Referenced by `in6_purgeaddr()`, and `in6_update_ifa()`.

Here is the call graph for this function:



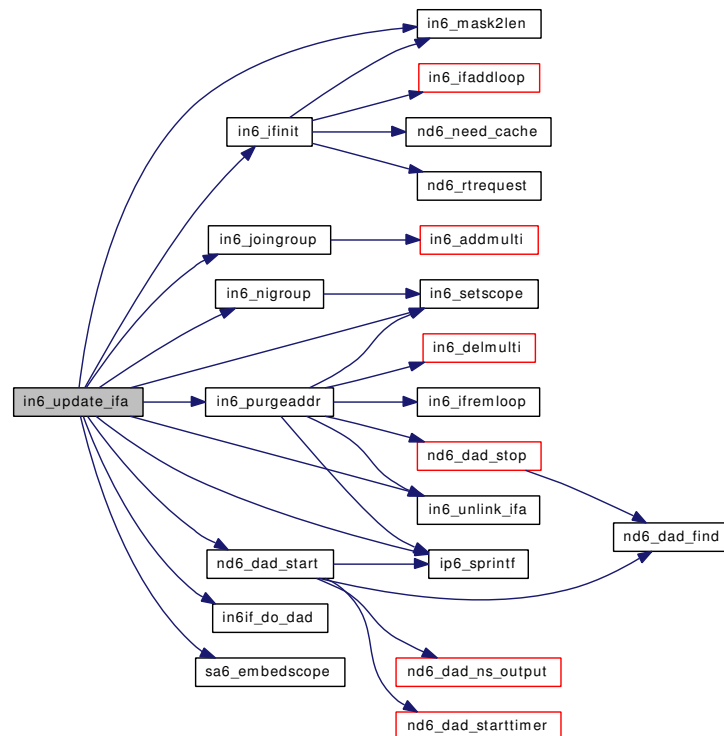
7.22.2.31 `int in6_update_ifa` (struct ifnet * *ifp*, struct `in6_aliasreq` * *ifra*, struct `in6_ifaddr` * *ia*, int *flags*)

Definition at line 798 of file `in6.c`.

References `hostnamelen`, `in6_multi_mship::i6mm_maddr`, `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia6_lifetime`, `in6_ifaddr::ia6_updatetime`, `in6_addrlifetime::ia6t_expire`, `in6_addrlifetime::ia6t_pltime`, `in6_addrlifetime::ia6t_preferred`, `in6_addrlifetime::ia6t_vltime`, `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_dstaddr`, `in6_ifaddr::ia_ifa`, `in6_ifaddr::ia_next`, `in6_ifaddr::ia_prefixmask`, `in6_aliasreq::ifra_addr`, `in6_aliasreq::ifra_dstaddr`, `in6_aliasreq::ifra_flags`, `in6_aliasreq::ifra_lifetime`, `in6_aliasreq::ifra_prefixmask`, `IN6_ARE_ADDR_EQUAL`, `IN6_IFAUPDATE_DADDELAY`, `IN6_IFF_DEPRECATED`, `IN6_IFF_DUPLICATED`, `IN6_IFF_NODAD`, `IN6_IFF_TENTATIVE`, `in6_ifinit()`, `in6_joyngroup()`, `in6_mask2len()`, `in6_nigroup()`, `in6_purgeaddr()`, `in6_setscope()`, `in6_unlink_ifa()`, `in6addr_linklocal_allnodes`, `in6addr_nodelocal_allnodes`, `in6if_do_dad()`, `in6_multi::in6m_state`, `in6_multi::in6m_timer`, `in6mask32`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `IPV6_ADDR_INT32_MLL`, `MAX_RTR_SOLICITATION_DELAY`, `MLD_REPORTPENDING`, `MLTMASK_LEN`, `nd6_dad_start()`, `ND6_INFINITE_LIFETIME`, `nd6log`, `sa6_embedscope()`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, `sockaddr_in6::sin6_len`, and `sockaddr_in6::sin6_scope_id`.

Referenced by `in6_control()`, `in6_ifadd()`, `in6_ifattach_linklocal()`, and `in6_ifattach_loopback()`.

Here is the call graph for this function:



7.22.2.32 int in6if_do_dad (struct ifnet * ifp)

Definition at line 2189 of file in6.c.

Referenced by in6_update_ifa().

7.22.2.33 struct in6_ifaddr* in6ifa_ifpforlinklocal (struct ifnet * ifp, int ignoreflags)

Definition at line 1854 of file in6.c.

References IFA_IN6, and IN6_IS_ADDR_LINKLOCAL.

Referenced by icmp6_redirect_output(), in6_ifadd(), in6_ifattach(), in6_ifattach_linklocal(), in6_lifaddr_ioctl(), mld6_sendpkt(), nd6_ns_input(), and nd6_prefix_onlink().

7.22.2.34 struct in6_ifaddr* in6ifa_ifpwithaddr (struct ifnet * ifp, struct in6_addr * addr)

Definition at line 1879 of file in6.c.

References IFA_IN6, and IN6_ARE_ADDR_EQUAL.

Referenced by in6_control(), in6_ifadd(), in6_ifattach(), nd6_na_input(), nd6_ns_input(), nd6_ns_output(), and nd6_output().

7.22.2.35 char* ip6_sprintf (char * ip6buf, const struct in6_addr * addr)

Definition at line 1901 of file in6.c.

Referenced by `add_m6fc()`, `del_m6fc()`, `expire_upcalls()`, `frag6_input()`, `gif_validate6()`, `icmp6_input()`, `icmp6_redirect_diag()`, `icmp6_redirect_input()`, `icmp6_reflect()`, `in6_ifadd()`, `in6_ifloop_request()`, `in6_purgeaddr()`, `in6_update_ifa()`, `ip6_forward()`, `ip6_input()`, `ip6_mforward()`, `ipsec_logsastr()`, `mld6_input()`, `nd6_dad_duplicated()`, `nd6_dad_ns_input()`, `nd6_dad_start()`, `nd6_dad_timer()`, `nd6_lookup()`, `nd6_na_input()`, `nd6_na_output()`, `nd6_ns_input()`, `nd6_ns_output()`, `nd6_prefix_offlink()`, `nd6_prefix_onlink()`, `nd6_prelist_add()`, `nd6_ra_input()`, `nd6_rs_input()`, `nd6_sysctl_prlist()`, `pfxlist_onlink_check()`, `phyint_send()`, `pim6_input()`, `prelist_remove()`, `prelist_update()`, `register_send()`, `sa6_recoverscope()`, `selectroute()`, and `udp6_input()`.

7.22.2.36 MALLOC_DEFINE (M_IP6MADDR, "in6_multi", "internet multicast address")

7.22.3 Variable Documentation

7.22.3.1 char `digits[]` = "0123456789abcdef" [static]

Definition at line 1899 of file `in6.c`.

7.22.3.2 int(*) `faithprefix_p(struct in6_addr *)`

Definition at line 131 of file `in6.c`.

Referenced by `icmp6_input()`, `in6_pcblookup_hash()`, `rip6_input()`, `sctp6_input()`, and `udp6_input()`.

7.22.3.3 struct `in6_multihead in6_multihead`

Definition at line 130 of file `in6.c`.

Referenced by `in6_addmulti()`, and `in6_ifdetach()`.

7.22.3.4 struct `in6_addr in6addr_any` = IN6ADDR_ANY_INIT

Definition at line 106 of file `in6.c`.

Referenced by `in6_pcbbind()`, `in6_pcbsetport()`, `rip6_disconnect()`, `udp6_abort()`, `udp6_close()`, and `udp6_disconnect()`.

7.22.3.5 struct `in6_addr in6addr_linklocal_allnodes`

Initial value:

```
IN6ADDR_LINKLOCAL_ALLNODES_INIT
```

Definition at line 110 of file `in6.c`.

Referenced by `in6_ifdetach()`, `in6_update_ifa()`, `mld6_input()`, `mld6_start_listening()`, `mld6_stop_listening()`, and `nd6_ns_input()`.

7.22.3.6 struct `in6_addr in6addr_linklocal_allrouters`

Initial value:

```
IN6ADDR_LINKLOCAL_ALLROUTERS_INIT
```

Definition at line 112 of file in6.c.

Referenced by mld6_stop_listening().

7.22.3.7 struct [in6_addr in6addr_loopback](#) = IN6ADDR_LOOPBACK_INIT

Definition at line 107 of file in6.c.

Referenced by in6_addrscope(), in6_ifattach(), in6_ifattach_loopback(), and in6_pcbladdr().

7.22.3.8 struct [in6_addr in6addr_nodelocal_allnodes](#)

Initial value:

```
IN6ADDR_NODELOCAL_ALLNODES_INIT
```

Definition at line 108 of file in6.c.

Referenced by in6_update_ifa().

7.22.3.9 struct [in6_addr in6mask0](#) = IN6MASK0

Definition at line 115 of file in6.c.

7.22.3.10 struct [in6_addr in6mask128](#) = IN6MASK128

Definition at line 119 of file in6.c.

Referenced by in6_ifattach_loopback(), and in6_ifloop_request().

7.22.3.11 struct [in6_addr in6mask32](#) = IN6MASK32

Definition at line 116 of file in6.c.

Referenced by in6_update_ifa().

7.22.3.12 struct [in6_addr in6mask64](#) = IN6MASK64

Definition at line 117 of file in6.c.

Referenced by in6_ifattach_linklocal().

7.22.3.13 struct [in6_addr in6mask96](#) = IN6MASK96

Definition at line 118 of file in6.c.

7.22.3.14 struct [sockaddr_in6 sa6_any](#)

Initial value:

```
{ sizeof(sa6_any), AF_INET6, 0, 0, IN6ADDR_ANY_INIT, 0 }
```

Definition at line 121 of file in6.c.

Referenced by [in6_pcbnotify\(\)](#), [rip6_ctlinput\(\)](#), and [udp6_ctlinput\(\)](#).

7.23 /usr/src/sys/netinet6/in6.h File Reference

Data Structures

- struct [in6_addr](#)
- struct [sockaddr_in6](#)
- struct [ipv6_mreq](#)
- struct [in6_pktinfo](#)
- struct [ip6_mtuinfo](#)

Defines

- #define [__KAME__](#)
- #define [__KAME_VERSION](#) "FreeBSD"
- #define [s6_addr __u6_addr.__u6_addr8](#)
- #define [s6_addr8 __u6_addr.__u6_addr8](#)
- #define [s6_addr16 __u6_addr.__u6_addr16](#)
- #define [s6_addr32 __u6_addr.__u6_addr32](#)
- #define [INET6_ADDRSTRLEN](#) 46
- #define [IN6MASK0](#) { { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 } }
- #define [IN6MASK32](#)
- #define [IN6MASK64](#)
- #define [IN6MASK96](#)
- #define [IN6MASK128](#)
- #define [IPV6_ADDR_INT32_ONE](#) 1
- #define [IPV6_ADDR_INT32_TWO](#) 2
- #define [IPV6_ADDR_INT32_MNL](#) 0xff010000
- #define [IPV6_ADDR_INT32_MLL](#) 0xff020000
- #define [IPV6_ADDR_INT32_SMP](#) 0x0000ffff
- #define [IPV6_ADDR_INT16_ULL](#) 0xfe80
- #define [IPV6_ADDR_INT16_USL](#) 0xfec0
- #define [IPV6_ADDR_INT16_MLL](#) 0xff02
- #define [IN6_ARE_ADDR_EQUAL\(a, b\)](#) (bcmp(&(a) → s6_addr[0], &(b) → s6_addr[0], sizeof(struct in6_addr)) == 0)
- #define [IN6_IS_ADDR_UNSPECIFIED\(a\)](#)
- #define [IN6_IS_ADDR_LOOPBACK\(a\)](#)
- #define [IN6_IS_ADDR_V4COMPAT\(a\)](#)
- #define [IN6_IS_ADDR_V4MAPPED\(a\)](#)
- #define [IPV6_ADDR_SCOPE_NODELOCAL](#) 0x01
- #define [IPV6_ADDR_SCOPE_INTFACELOCAL](#) 0x01
- #define [IPV6_ADDR_SCOPE_LINKLOCAL](#) 0x02
- #define [IPV6_ADDR_SCOPE_SITELOCAL](#) 0x05
- #define [IPV6_ADDR_SCOPE_ORGLOCAL](#) 0x08
- #define [IPV6_ADDR_SCOPE_GLOBAL](#) 0x0e
- #define [IN6_IS_ADDR_LINKLOCAL\(a\)](#) (((a) → s6_addr[0] == 0xfe) && (((a) → s6_addr[1] & 0xc0) == 0x80))
- #define [IN6_IS_ADDR_SITELOCAL\(a\)](#) (((a) → s6_addr[0] == 0xfe) && (((a) → s6_addr[1] & 0xc0) == 0xc0))
- #define [IN6_IS_ADDR_MULTICAST\(a\)](#) ((a) → s6_addr[0] == 0xff)
- #define [IPV6_ADDR_MC_SCOPE\(a\)](#) ((a) → s6_addr[1] & 0x0f)

- #define IN6_IS_ADDR_MC_NODELOCAL(a)
- #define IN6_IS_ADDR_MC_INTFACELOCAL(a)
- #define IN6_IS_ADDR_MC_LINKLOCAL(a)
- #define IN6_IS_ADDR_MC_SITELOCAL(a)
- #define IN6_IS_ADDR_MC_ORGLOCAL(a)
- #define IN6_IS_ADDR_MC_GLOBAL(a)
- #define IN6_IS_SCOPE_LINKLOCAL(a)
- #define IFA6_IS_DEPRECATED(a)
- #define IFA6_IS_INVALID(a)
- #define IPV6_SOCKOPT_RESERVED1 3
- #define IPV6_UNICAST_HOPS 4
- #define IPV6_MULTICAST_IF 9
- #define IPV6_MULTICAST_HOPS 10
- #define IPV6_MULTICAST_LOOP 11
- #define IPV6_JOIN_GROUP 12
- #define IPV6_LEAVE_GROUP 13
- #define IPV6_PORTRANGE 14
- #define ICMP6_FILTER 18
- #define IPV6_2292PKTINFO 19
- #define IPV6_2292HOPLIMIT 20
- #define IPV6_2292NEXTHOP 21
- #define IPV6_2292HOPOPTS 22
- #define IPV6_2292DSTOPTS 23
- #define IPV6_2292RTHDR 24
- #define IPV6_2292PKTOPTIONS 25
- #define IPV6_CHECKSUM 26
- #define IPV6_V6ONLY 27
- #define IPV6_IPSEC_POLICY 28
- #define IPV6_FAITH 29
- #define IPV6_FW_ADD 30
- #define IPV6_FW_DEL 31
- #define IPV6_FW_FLUSH 32
- #define IPV6_FW_ZERO 33
- #define IPV6_FW_GET 34
- #define IPV6_RTHDRDSTOPTS 35
- #define IPV6_RECVPKTINFO 36
- #define IPV6_RECVHOPLIMIT 37
- #define IPV6_RECVRTHDR 38
- #define IPV6_RECVHOPOPTS 39
- #define IPV6_RECVDSTOPTS 40
- #define IPV6_RECVRTHDRDSTOPTS 41
- #define IPV6_USE_MIN_MTU 42
- #define IPV6_RECVPATHMTU 43
- #define IPV6_PATHMTU 44
- #define IPV6_PKTINFO 46
- #define IPV6_HOPLIMIT 47
- #define IPV6_NEXTHOP 48
- #define IPV6_HOPOPTS 49
- #define IPV6_DSTOPTS 50
- #define IPV6_RTHDR 51

- #define [IPV6_RECVTCLASS](#) 57
- #define [IPV6_AUTOFLOWLABEL](#) 59
- #define [IPV6_TCLASS](#) 61
- #define [IPV6_DONTFRAG](#) 62
- #define [IPV6_PREFER_TEMPADDR](#) 63
- #define [IPV6_RTHDR_LOOSE](#) 0
- #define [IPV6_RTHDR_STRICT](#) 1
- #define [IPV6_RTHDR_TYPE_0](#) 0
- #define [IPV6_DEFAULT_MULTICAST_HOPS](#) 1
- #define [IPV6_DEFAULT_MULTICAST_LOOP](#) 1
- #define [IPV6_PORTRANGE_DEFAULT](#) 0
- #define [IPV6_PORTRANGE_HIGH](#) 1
- #define [IPV6_PORTRANGE_LOW](#) 2
- #define [M_AUTHIPHDR](#) M_PROTO2
- #define [M_DECRYPTED](#) M_PROTO3
- #define [M_LOOP](#) M_PROTO4
- #define [M_AUTHIPDGM](#) M_PROTO5
- #define [satosin6\(sa\)](#) ((struct [sockaddr_in6](#) *) (sa))
- #define [sin6tosa\(sin6\)](#) ((struct [sockaddr](#) *) (sin6))
- #define [ifatoia6\(ifa\)](#) ((struct [in6_ifaddr](#) *) (ifa))

Typedefs

- typedef [__size_t](#) [size_t](#)
- typedef [__socklen_t](#) [socklen_t](#)

Functions

- int [in6_cksum](#) [__P](#) ((struct [mbuf](#) *, [u_int8_t](#), [u_int32_t](#), [u_int32_t](#))
- int [in6_localaddr](#) [__P](#) ((struct [in6_addr](#) *)
- [in6_ifaddr](#) *[in6_ifawithifp](#) [__P](#) ((struct [ifnet](#) *, struct [in6_addr](#) *)
- void [in6_if_up](#) [__P](#) ((struct [ifnet](#) *)
- void [in6_sin6_2_sin](#) [__P](#) ((struct [sockaddr_in](#) *[sin](#), struct [sockaddr_in6](#) *[sin6](#))
- void [in6_sin6_2_sin_in_sock](#) [__P](#) ((struct [sockaddr](#) *[nam](#))
- void [in6_sin_2_v4mapsin6_in_sock](#) [__P](#) ((struct [sockaddr](#) **[nam](#))
- void [addrsel_policy_init](#) [__P](#) ((void))

Variables

- [sockaddr_in6](#) [sa6_any](#)
- [in6_addr](#) [in6mask0](#)
- [in6_addr](#) [in6mask32](#)
- [in6_addr](#) [in6mask64](#)
- [in6_addr](#) [in6mask96](#)
- [in6_addr](#) [in6mask128](#)
- [in6_addr](#) [in6addr_any](#)
- [in6_addr](#) [in6addr_loopback](#)
- [u_char](#) [ip6_protox](#) []
- int(*) [faithprefix_p](#) (struct [in6_addr](#) *)

7.23.1 Define Documentation

7.23.1.1 #define __KAME__

Definition at line 76 of file in6.h.

7.23.1.2 #define __KAME_VERSION "FreeBSD"

Definition at line 77 of file in6.h.

7.23.1.3 #define ICMP6_FILTER 18

Definition at line 394 of file in6.h.

Referenced by icmp6_ctloutput().

7.23.1.4 #define IFA6_IS_DEPRECATED(a)

Value:

```
((a)->ia6_lifetime.ia6t_pltime != ND6_INFINITE_LIFETIME && \
 (u_int32_t)((time_second - (a)->ia6_updatetime) > \
 (a)->ia6_lifetime.ia6t_pltime)
```

Definition at line 354 of file in6.h.

Referenced by in6_selectsrc(), nd6_timer(), and regen_tmpaddr().

7.23.1.5 #define IFA6_IS_INVALID(a)

Value:

```
((a)->ia6_lifetime.ia6t_vltime != ND6_INFINITE_LIFETIME && \
 (u_int32_t)((time_second - (a)->ia6_updatetime) > \
 (a)->ia6_lifetime.ia6t_vltime)
```

Definition at line 358 of file in6.h.

Referenced by nd6_timer().

7.23.1.6 #define ifatoia6(ifa) ((struct in6_ifaddr *) (ifa))

Definition at line 609 of file in6.h.

7.23.1.7 #define IN6_ARE_ADDR_EQUAL(a, b) (bcmp(&(a) -> s6_addr[0], &(b) -> s6_addr[0], sizeof(struct in6_addr)) == 0)

Definition at line 221 of file in6.h.

Referenced by add_addrsel_policyent(), add_m6fc(), defrouter_lookup(), del_m6fc(), delete_addrsel_policyent(), frag6_input(), gif_validate6(), icmp6_rip6_input(), in6_addroute(), in6_gif_output(), in6_ifremloop(), in6_is_addr_deprecated(), in6_lifaddr_ioctl(), in6_pcblookup_hash(), in6_pcblookup_local(),

in6_pcbnotify(), in6_selectsrc(), in6_update_ifa(), in6ifa_ifpwithaddr(), ip6_forward(), ip6_getpmtu(), ip6_input(), ip6_mforward(), ip6_output(), ip6_setmoptions(), mld6_input(), mld6_start_listening(), mld6_stop_listening(), nd6_dad_duplicated(), nd6_ns_input(), ni6_addrs(), ni6_input(), rip6_input(), rt6_deleteroute(), sctp6_input(), selectroute(), and udp6_input().

7.23.1.8 #define IN6_IS_ADDR_LINKLOCAL(a) (((a) → s6_addr[0] == 0xfe) && (((a) → s6_addr[1] & 0xc0) == 0x80))

Definition at line 290 of file in6.h.

Referenced by icmp6_redirect_input(), icmp6_redirect_output(), in6_ifdetach(), in6_localaddr(), in6ifa_ifpforlinklocal(), mld6_input(), nd6_dad_duplicated(), nd6_ioctl(), nd6_is_new_addr_neighbor(), nd6_ra_input(), pfxlist_onlink_check(), and rt6_flush().

7.23.1.9 #define IN6_IS_ADDR_LOOPBACK(a)

Value:

```
((*(const u_int32_t *) (const void *) (&(a)->s6_addr[0]) == 0) && \
  (*(const u_int32_t *) (const void *) (&(a)->s6_addr[4]) == 0) && \
  (*(const u_int32_t *) (const void *) (&(a)->s6_addr[8]) == 0) && \
  (*(const u_int32_t *) (const void *) (&(a)->s6_addr[12]) == ntohl(1)))
```

Definition at line 242 of file in6.h.

Referenced by in6_localaddr(), in6_setscope(), scope6_addr2default(), and selectroute().

7.23.1.10 #define IN6_IS_ADDR_MC_GLOBAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
  (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_GLOBAL))
```

Definition at line 325 of file in6.h.

7.23.1.11 #define IN6_IS_ADDR_MC_INTFACELOCAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
  (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_INTFACELOCAL))
```

Definition at line 313 of file in6.h.

Referenced by in6_clearscope(), in6_setscope(), ip6_input(), ip6_mforward(), ip6_output(), sa6_embedscope(), and sa6_recoverscope().

7.23.1.12 #define IN6_IS_ADDR_MC_LINKLOCAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
 (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_LINKLOCAL))
```

Definition at line 316 of file in6.h.

Referenced by ip6_mforward(), and ni6_input().

7.23.1.13 #define IN6_IS_ADDR_MC_NODELOCAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
 (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_NODELOCAL))
```

Definition at line 310 of file in6.h.

7.23.1.14 #define IN6_IS_ADDR_MC_ORGLOCAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
 (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_ORGLOCAL))
```

Definition at line 322 of file in6.h.

7.23.1.15 #define IN6_IS_ADDR_MC_SITELOCAL(a)

Value:

```
(IN6_IS_ADDR_MULTICAST(a) && \
 (IPV6_ADDR_MC_SCOPE(a) == IPV6_ADDR_SCOPE_SITELOCAL))
```

Definition at line 319 of file in6.h.

7.23.1.16 #define IN6_IS_ADDR_MULTICAST(a) ((a) → s6_addr[0] == 0xff)

Definition at line 298 of file in6.h.

Referenced by icmp6_error(), icmp6_redirect_input(), icmp6_redirect_output(), icmp6_reflect(), in6_addroute(), in6_pcbbind(), ip6_forward(), ip6_input(), ip6_output(), ip6_rthdr0(), ip6_setmoptions(), ip6_setpktopt(), ip6_unknown_opt(), mld6_input(), nd6_na_input(), nd6_na_output(), nd6_ns_input(), nd6_ns_output(), nd6_output(), nd6_ra_input(), ni6_input(), pim6_input(), sctp6_input(), selectroute(), and udp6_input().

7.23.1.17 #define IN6_IS_ADDR_SITELOCAL(a) (((a) → s6_addr[0] == 0xfe) && (((a) → s6_addr[1] & 0xc0) == 0xc0))

Definition at line 292 of file in6.h.

7.23.1.18 #define IN6_IS_ADDR_UNSPECIFIED(a)**Value:**

```
((*(const u_int32_t *) (const void *) (&(a)->s6_addr[0]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[4]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[8]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[12]) == 0))
```

Definition at line 233 of file in6.h.

Referenced by icmp6_error(), icmp6_rip6_input(), in6_gif_output(), in6_pcbbind(), in6_pcbconnect(), in6_pcblladdr(), in6_pcblookup_hash(), in6_pcblookup_local(), in6_pcbnotify(), in6_selectthlim(), in6_selectsrc(), ip6_ctloutput(), ip6_forward(), ip6_input(), ip6_mforward(), ip6_output(), ip6_rthdr0(), ip6_setmoptions(), ip6_setpktopt(), mld6_input(), nd6_cache_lladdr(), nd6_na_output(), nd6_ns_input(), nd6_rs_input(), rip6_bind(), rip6_input(), sctp6_bind(), udp6_abort(), udp6_bind(), udp6_close(), udp6_connect(), udp6_disconnect(), udp6_input(), udp6_output(), and udp6_send().

7.23.1.19 #define IN6_IS_ADDR_V4COMPAT(a)**Value:**

```
((*(const u_int32_t *) (const void *) (&(a)->s6_addr[0]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[4]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[8]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[12]) != 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[12]) != ntohl(1)))
```

Definition at line 251 of file in6.h.

Referenced by ip6_input(), and ip6_rthdr0().

7.23.1.20 #define IN6_IS_ADDR_V4MAPPED(a)**Value:**

```
((*(const u_int32_t *) (const void *) (&(a)->s6_addr[0]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[4]) == 0) && \
 (*(const u_int32_t *) (const void *) (&(a)->s6_addr[8]) == ntohl(0x0000ffff)))
```

Definition at line 261 of file in6.h.

Referenced by ip6_input(), ip6_rthdr0(), sctp6_bind(), sctp6_connect(), sctp6_send(), udp6_bind(), udp6_connect(), udp6_output(), and udp6_send().

7.23.1.21 #define IN6_IS_SCOPE_LINKLOCAL(a)**Value:**

```
((IN6_IS_ADDR_LINKLOCAL(a)) || \
 (IN6_IS_ADDR_MC_LINKLOCAL(a)))
```

Definition at line 350 of file in6.h.

Referenced by in6_clearscope(), in6_setscope(), sa6_embedscope(), and sa6_recoverscope().

7.23.1.22 #define IN6MASK0 {{{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }}}}

Definition at line 136 of file in6.h.

7.23.1.23 #define IN6MASK128**Value:**

```
{{{ 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, \
      0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff }}}}
```

Definition at line 143 of file in6.h.

7.23.1.24 #define IN6MASK32**Value:**

```
{{{ 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, \
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }}}}
```

Definition at line 137 of file in6.h.

7.23.1.25 #define IN6MASK64**Value:**

```
{{{ 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, \
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }}}}
```

Definition at line 139 of file in6.h.

7.23.1.26 #define IN6MASK96**Value:**

```
{{{ 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, \
      0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00 }}}}
```

Definition at line 141 of file in6.h.

7.23.1.27 #define INET6_ADDRSTRLEN 46

Definition at line 110 of file in6.h.

Referenced by add_m6fc(), del_m6fc(), expire_upcalls(), frag6_input(), gif_validate6(), icmp6_input(), icmp6_redirect_diag(), icmp6_redirect_input(), icmp6_reflect(), in6_ifadd(), in6_ifloop_request(), in6_purgeaddr(), in6_update_ifa(), ip6_forward(), ip6_input(), ip6_mforward(), ipsec_logsastr(), mld6_input(), nd6_dad_duplicated(), nd6_dad_ns_input(), nd6_dad_start(), nd6_dad_timer(), nd6_lookup(), nd6_na_input(), nd6_na_output(), nd6_ns_input(), nd6_ns_output(), nd6_prefix_offlink(), nd6_prefix_onlink(), nd6_prelist_add(), nd6_ra_input(), nd6_rs_input(), nd6_sysctl_prlist(), pfxlist_onlink_check(), phyint_send(), pim6_input(), prelist_remove(), prelist_update(), register_send(), sa6_recoverscope(), selectroute(), and udp6_input().

7.23.1.28 #define IPV6_2292DSTOPTS 23

Definition at line 401 of file in6.h.

Referenced by ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.29 #define IPV6_2292HOPLIMIT 20

Definition at line 398 of file in6.h.

Referenced by ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.30 #define IPV6_2292HOPOPTS 22

Definition at line 400 of file in6.h.

Referenced by ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.31 #define IPV6_2292NEXTHOP 21

Definition at line 399 of file in6.h.

Referenced by ip6_setpktopt().

7.23.1.32 #define IPV6_2292PKTINFO 19

Definition at line 397 of file in6.h.

Referenced by ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.33 #define IPV6_2292PKTOPTIONS 25

Definition at line 403 of file in6.h.

Referenced by ip6_ctloutput(), and ip6_setpktopt().

7.23.1.34 #define IPV6_2292RTHDR 24

Definition at line 402 of file in6.h.

Referenced by ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.35 #define IPV6_ADDR_INT16_MLL 0xff02

Definition at line 169 of file in6.h.

Referenced by in6_nigroup(), nd6_na_output(), nd6_ns_input(), and nd6_ns_output().

7.23.1.36 #define IPV6_ADDR_INT16_ULL 0xfe80

Definition at line 167 of file in6.h.

7.23.1.37 #define IPV6_ADDR_INT16_USL 0xfec0

Definition at line 168 of file in6.h.

7.23.1.38 #define IPV6_ADDR_INT32_MLL 0xff020000

Definition at line 165 of file in6.h.

Referenced by in6_purgeaddr(), and in6_update_ifa().

7.23.1.39 #define IPV6_ADDR_INT32_MNL 0xff010000

Definition at line 164 of file in6.h.

7.23.1.40 #define IPV6_ADDR_INT32_ONE 1

Definition at line 162 of file in6.h.

Referenced by nd6_na_output(), nd6_ns_input(), and nd6_ns_output().

7.23.1.41 #define IPV6_ADDR_INT32_SMP 0x0000ffff

Definition at line 166 of file in6.h.

Referenced by in6_sin_2_v4mapsin6().

7.23.1.42 #define IPV6_ADDR_INT32_TWO 2

Definition at line 163 of file in6.h.

7.23.1.43 #define IPV6_ADDR_MC_SCOPE(a) ((a) → s6_addr[1] & 0x0f)

Definition at line 301 of file in6.h.

Referenced by mld6_input(), mld6_start_listening(), and mld6_stop_listening().

7.23.1.44 #define IPV6_ADDR_SCOPE_GLOBAL 0x0e

Definition at line 276 of file in6.h.

Referenced by in6_addrscope(), ni6_addrs(), and ni6_store_addrs().

7.23.1.45 #define IPV6_ADDR_SCOPE_INTFACELOCAL 0x01

Definition at line 272 of file in6.h.

Referenced by in6_addrscope(), in6_setscope(), mld6_stop_listening(), scope6_ifattach(), scope6_set(), and scope6_setdefault().

7.23.1.46 #define IPV6_ADDR_SCOPE_LINKLOCAL 0x02

Definition at line 273 of file in6.h.

Referenced by in6_addrscope(), in6_setscope(), mld6_input(), mld6_start_listening(), ni6_addrs(), ni6_store_addrs(), scope6_ifattach(), scope6_set(), and scope6_setdefault().

7.23.1.47 #define IPV6_ADDR_SCOPE_NODELOCAL 0x01

Definition at line 271 of file in6.h.

7.23.1.48 #define IPV6_ADDR_SCOPE_ORGLOCAL 0x08

Definition at line 275 of file in6.h.

Referenced by in6_setscope(), and scope6_ifattach().

7.23.1.49 #define IPV6_ADDR_SCOPE_SITELOCAL 0x05

Definition at line 274 of file in6.h.

Referenced by in6_addrscope(), in6_setscope(), ni6_addrs(), ni6_store_addrs(), and scope6_ifattach().

7.23.1.50 #define IPV6_AUTOFLOWLABEL 59

Definition at line 461 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.51 #define IPV6_CHECKSUM 26

Definition at line 406 of file in6.h.

Referenced by ip6_raw_ctloutput(), and rip6_ctloutput().

7.23.1.52 #define IPV6_DEFAULT_MULTICAST_HOPS 1

Definition at line 479 of file in6.h.

7.23.1.53 #define IPV6_DEFAULT_MULTICAST_LOOP 1

Definition at line 480 of file in6.h.

Referenced by ip6_setmoptions().

7.23.1.54 #define IPV6_DONTFRAG 62

Definition at line 464 of file in6.h.

Referenced by ip6_ctloutput(), ip6_getpcbopt(), and ip6_setpktopt().

7.23.1.55 #define IPV6_DSTOPTS 50

Definition at line 452 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.56 #define IPV6_FAITH 29

Definition at line 415 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.57 #define IPV6_FW_ADD 30

Definition at line 418 of file in6.h.

7.23.1.58 #define IPV6_FW_DEL 31

Definition at line 419 of file in6.h.

7.23.1.59 #define IPV6_FW_FLUSH 32

Definition at line 420 of file in6.h.

7.23.1.60 #define IPV6_FW_GET 34

Definition at line 422 of file in6.h.

7.23.1.61 #define IPV6_FW_ZERO 33

Definition at line 421 of file in6.h.

7.23.1.62 #define IPV6_HOPLIMIT 47

Definition at line 449 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.63 #define IPV6_HOPOPTS 49

Definition at line 451 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.64 #define IPV6_IPSEC_POLICY 28

Definition at line 413 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.65 #define IPV6_JOIN_GROUP 12

Definition at line 391 of file in6.h.

Referenced by ip6_ctloutput(), and ip6_setmoptions().

7.23.1.66 #define IPV6_LEAVE_GROUP 13

Definition at line 392 of file in6.h.

Referenced by ip6_ctloutput(), and ip6_setmoptions().

7.23.1.67 #define IPV6_MULTICAST_HOPS 10

Definition at line 389 of file in6.h.

Referenced by ip6_ctloutput(), ip6_getmoptions(), and ip6_setmoptions().

7.23.1.68 #define IPV6_MULTICAST_IF 9

Definition at line 388 of file in6.h.

Referenced by ip6_ctloutput(), ip6_getmoptions(), and ip6_setmoptions().

7.23.1.69 #define IPV6_MULTICAST_LOOP 11

Definition at line 390 of file in6.h.

Referenced by ip6_ctloutput(), ip6_getmoptions(), and ip6_setmoptions().

7.23.1.70 #define IPV6_NEXTHOP 48

Definition at line 450 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), and ip6_setpktopt().

7.23.1.71 #define IPV6_PATHMTU 44

Definition at line 440 of file in6.h.

Referenced by ip6_ctloutput(), and ip6_notify_pmtu().

7.23.1.72 #define IPV6_PKTINFO 46

Definition at line 448 of file in6.h.

Referenced by `ip6_clearpktopts()`, `ip6_ctloutput()`, `ip6_getpcbopt()`, `ip6_savecontrol()`, and `ip6_setpktopt()`.

7.23.1.73 #define IPV6_PORTRANGE 14

Definition at line 393 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.74 #define IPV6_PORTRANGE_DEFAULT 0

Definition at line 510 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.75 #define IPV6_PORTRANGE_HIGH 1

Definition at line 511 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.76 #define IPV6_PORTRANGE_LOW 2

Definition at line 512 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.77 #define IPV6_PREFER_TEMPADDR 63

Definition at line 466 of file `in6.h`.

Referenced by `ip6_ctloutput()`, `ip6_getpcbopt()`, and `ip6_setpktopt()`.

7.23.1.78 #define IPV6_RECVDSTOPTS 40

Definition at line 432 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.79 #define IPV6_RECVHOPLIMIT 37

Definition at line 429 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.80 #define IPV6_RECVHOPOPTS 39

Definition at line 431 of file `in6.h`.

Referenced by `ip6_ctloutput()`.

7.23.1.81 #define IPV6_RECVPATHMTU 43

Definition at line 438 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.82 #define IPV6_RECVPKTINFO 36

Definition at line 428 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.83 #define IPV6_RECVRTHDR 38

Definition at line 430 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.84 #define IPV6_RECVRTHDRDSTOPTS 41

Definition at line 434 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.85 #define IPV6_RECVTCLASS 57

Definition at line 459 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.86 #define IPV6_RTHDR 51

Definition at line 453 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.87 #define IPV6_RTHDR_LOOSE 0

Definition at line 472 of file in6.h.

7.23.1.88 #define IPV6_RTHDR_STRICT 1

Definition at line 473 of file in6.h.

7.23.1.89 #define IPV6_RTHDR_TYPE_0 0

Definition at line 474 of file in6.h.

Referenced by icmp6_notify_error(), ip6_output(), ip6_setpktopt(), and route6_input().

7.23.1.90 #define IPV6_RTHDRDSTOPTS 35

Definition at line 426 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), and ip6_setpktopt().

7.23.1.91 #define IPV6_SOCKOPT_RESERVED1 3

Definition at line 386 of file in6.h.

7.23.1.92 #define IPV6_TCLASS 61

Definition at line 463 of file in6.h.

Referenced by ip6_clearpktopts(), ip6_ctloutput(), ip6_getpcbopt(), ip6_savecontrol(), and ip6_setpktopt().

7.23.1.93 #define IPV6_UNICAST_HOPS 4

Definition at line 387 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.94 #define IPV6_USE_MIN_MTU 42

Definition at line 437 of file in6.h.

Referenced by ip6_ctloutput(), ip6_getpcbopt(), and ip6_setpktopt().

7.23.1.95 #define IPV6_V6ONLY 27

Definition at line 407 of file in6.h.

Referenced by ip6_ctloutput().

7.23.1.96 #define M_AUTHIPDGM M_PROTO5

Definition at line 586 of file in6.h.

Referenced by ip6_input(), ipsec_in_reject(), and prelist_update().

7.23.1.97 #define M_AUTHIPHDR M_PROTO2

Definition at line 583 of file in6.h.

Referenced by ip6_input(), ipsec_in_reject(), and prelist_update().

7.23.1.98 #define M_DECRYPTED M_PROTO3

Definition at line 584 of file in6.h.

Referenced by icmp6_error(), icmp6_redirect_output(), and ipsec_in_reject().

7.23.1.99 #define M_LOOP M_PROTO4

Definition at line 585 of file in6.h.

Referenced by ip6_input(), ip6_mdq(), ip6_output(), and mld6_input().

7.23.1.100 #define s6_addr __u6_addr.__u6_addr8

Definition at line 103 of file in6.h.

7.23.1.101 #define s6_addr16 __u6_addr.__u6_addr16

Definition at line 106 of file in6.h.

7.23.1.102 #define s6_addr32 __u6_addr.__u6_addr32

Definition at line 107 of file in6.h.

7.23.1.103 #define s6_addr8 __u6_addr.__u6_addr8

Definition at line 105 of file in6.h.

7.23.1.104 #define satosin6(sa) ((struct sockaddr_in6 *)(sa))

Definition at line 607 of file in6.h.

Referenced by in6_addroute(), in6_ifdetach(), and selectroute().

7.23.1.105 #define sin6tosa(sin6) ((struct sockaddr *)(sin6))

Definition at line 608 of file in6.h.

Referenced by sctp6_getcred().

7.23.2 Typedef Documentation**7.23.2.1 typedef __size_t size_t**

Definition at line 615 of file in6.h.

7.23.2.2 typedef __socklen_t socklen_t

Definition at line 620 of file in6.h.

7.23.3 Function Documentation

7.23.3.1 void `addrsel_policy_init` __P ((void))

7.23.3.2 void `in6_sin_2_v4mapsin6_in_sock` __P ((struct `sockaddr **nam`))

7.23.3.3 void `in6_sin6_2_sin_in_sock` __P ((struct `sockaddr *nam`))

7.23.3.4 void `in6_sin_2_v4mapsin6` __P ((struct `sockaddr_in *sin`, struct `sockaddr_in6 *sin6`))

7.23.3.5 void `scope6_setdefault` __P ((struct `ifnet *`))

7.23.3.6 struct `in6_ifaddr*` `in6_ifawithifp` __P ((struct `ifnet *`, struct `in6_addr *`))

7.23.3.7 int `in6_clearscope` __P ((struct `in6_addr *`))

7.23.3.8 int `in6_cksum` __P ((struct `mbuf *`, `u_int8_t`, `u_int32_t`, `u_int32_t`))

7.23.4 Variable Documentation

7.23.4.1 int(*) `faithprefix_p`(struct `in6_addr *`)

Definition at line 131 of file `in6.c`.

Referenced by `icmp6_input()`, `in6_pcblookup_hash()`, `rip6_input()`, `sctp6_input()`, and `udp6_input()`.

7.23.4.2 struct `in6_addr` `in6addr_any`

Definition at line 106 of file `in6.c`.

Referenced by `in6_pcbbind()`, `in6_pcbsetport()`, `rip6_disconnect()`, `udp6_abort()`, `udp6_close()`, and `udp6_disconnect()`.

7.23.4.3 struct `in6_addr` `in6addr_loopback`

Definition at line 107 of file `in6.c`.

Referenced by `in6_addrscope()`, `in6_ifattach()`, `in6_ifattach_loopback()`, and `in6_pcbldaddr()`.

7.23.4.4 struct `in6_addr` `in6mask0`

Definition at line 115 of file `in6.c`.

7.23.4.5 struct `in6_addr` `in6mask128`

Definition at line 119 of file `in6.c`.

Referenced by `in6_ifattach_loopback()`, and `in6_ifloop_request()`.

7.23.4.6 struct `in6_addr` `in6mask32`

Definition at line 116 of file `in6.c`.

Referenced by `in6_update_ifa()`.

7.23.4.7 struct `in6_addr in6mask64`

Definition at line 117 of file `in6.c`.

Referenced by `in6_ifattach_linklocal()`.

7.23.4.8 struct `in6_addr in6mask96`

Definition at line 118 of file `in6.c`.

7.23.4.9 u_char `ip6_protox[]`

Definition at line 121 of file `ip6_input.c`.

Referenced by `icmp6_notify_error()`, `ip6_init()`, and `ip6_input()`.

7.23.4.10 struct `sockaddr_in6 sa6_any`

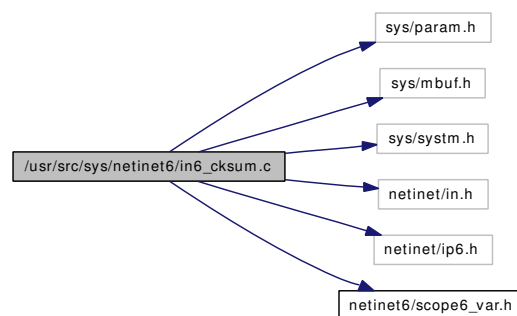
Definition at line 121 of file `in6.c`.

Referenced by `in6_pcnotify()`, `rip6_ctlinput()`, and `udp6_ctlinput()`.

7.24 /usr/src/sys/netinet6/in6_cksum.c File Reference

```
#include <sys/param.h>
#include <sys/mbuf.h>
#include <sys/system.h>
#include <netinet/in.h>
#include <netinet/ip6.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for in6_cksum.c:



Defines

- #define `ADDCARRY(x)` (`x > 65535 ? x -= 65535 : x`)
- #define `REDUCE {l_util.l = sum; sum = l_util.s[0] + l_util.s[1]; ADDCARRY(sum);}`

Functions

- int `in6_cksum` (struct mbuf *m, u_int8_t nxt, u_int32_t off, u_int32_t len)

7.24.1 Define Documentation

7.24.1.1 #define `ADDCARRY(x)` (`x > 65535 ? x -= 65535 : x`)

Definition at line 78 of file `in6_cksum.c`.

7.24.1.2 #define `REDUCE {l_util.l = sum; sum = l_util.s[0] + l_util.s[1]; ADDCARRY(sum);}`

Definition at line 79 of file `in6_cksum.c`.

Referenced by `in6_cksum()`.

7.24.2 Function Documentation

7.24.2.1 int in6_cksum (struct mbuf * m, u_int8_t nxt, u_int32_t off, u_int32_t len)

Definition at line 89 of file in6_cksum.c.

References in6_clearscope(), and REDUCE.

Referenced by icmp6_input(), icmp6_redirect_output(), icmp6_reflect(), mld6_sendpkt(), nd6_na_output(), nd6_ns_output(), pim6_input(), rip6_input(), rip6_output(), udp6_input(), and udp6_output().

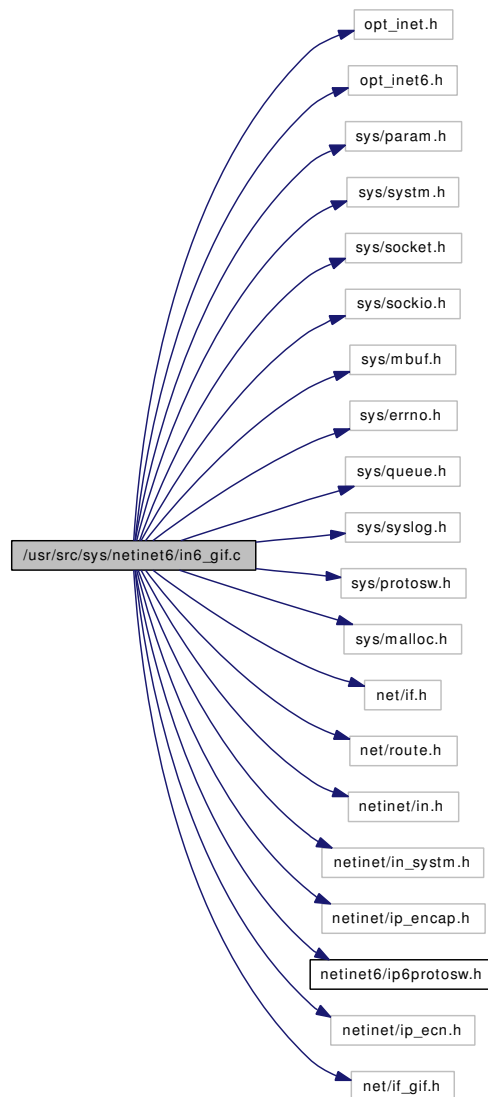
Here is the call graph for this function:



7.25 /usr/src/sys/netinet6/in6_gif.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/socket.h>  
#include <sys/sockio.h>  
#include <sys/mbuf.h>  
#include <sys/errno.h>  
#include <sys/queue.h>  
#include <sys/syslog.h>  
#include <sys/protosw.h>  
#include <sys/malloc.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/ip_encap.h>  
#include <netinet6/ip6protosw.h>  
#include <netinet/ip_ecn.h>  
#include <net/if_gif.h>
```

Include dependency graph for in6_gif.c:



Functions

- static int [gif_validate6](#) (const struct ip6_hdr *, struct gif_softc *, struct ifnet *)
- int [in6_gif_output](#) (struct ifnet *ifp, int family, struct mbuf *m)
- int [in6_gif_input](#) (struct mbuf **mp, int *offp, int proto)
- static int [gif_validate6](#) (struct ip6_hdr *ip6, struct gif_softc *sc, struct ifnet *ifp) const
- int [gif_encapcheck6](#) (struct mbuf *m, int off, int proto, void *arg) const
- int [in6_gif_attach](#) (struct gif_softc *sc)
- int [in6_gif_detach](#) (struct gif_softc *sc)

Variables

- domain [inet6domain](#)
- [ip6protosw](#) `in6_gif_protosw`

7.25.1 Function Documentation

7.25.1.1 `int gif_encapcheck6 (struct mbuf * m, int off, int proto, void * arg) const`

Definition at line 393 of file `in6_gif.c`.

References `gif_validate6()`.

Here is the call graph for this function:



7.25.1.2 `static int gif_validate6 (struct ip6_hdr * ip6, struct gif_softc * sc, struct ifnet * ifp) const` [static]

Definition at line 337 of file `in6_gif.c`.

References `IN6_ARE_ADDR_EQUAL`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `sin6`, and `sockaddr_in6::sin6_addr`.

Here is the call graph for this function:



7.25.1.3 `static int gif_validate6 (const struct ip6_hdr *, struct gif_softc *, struct ifnet *)` [static]

Referenced by `gif_encapcheck6()`.

7.25.1.4 `int in6_gif_attach (struct gif_softc * sc)`

Definition at line 414 of file `in6_gif.c`.

References `in6_gif_protosw`.

7.25.1.5 `int in6_gif_detach (struct gif_softc * sc)`

Definition at line 425 of file `in6_gif.c`.

7.25.1.6 `int in6_gif_input (struct mbuf ** mp, int * offp, int proto)`

Definition at line 246 of file `in6_gif.c`.

References `ip6_ecn_egress()`.

Here is the call graph for this function:

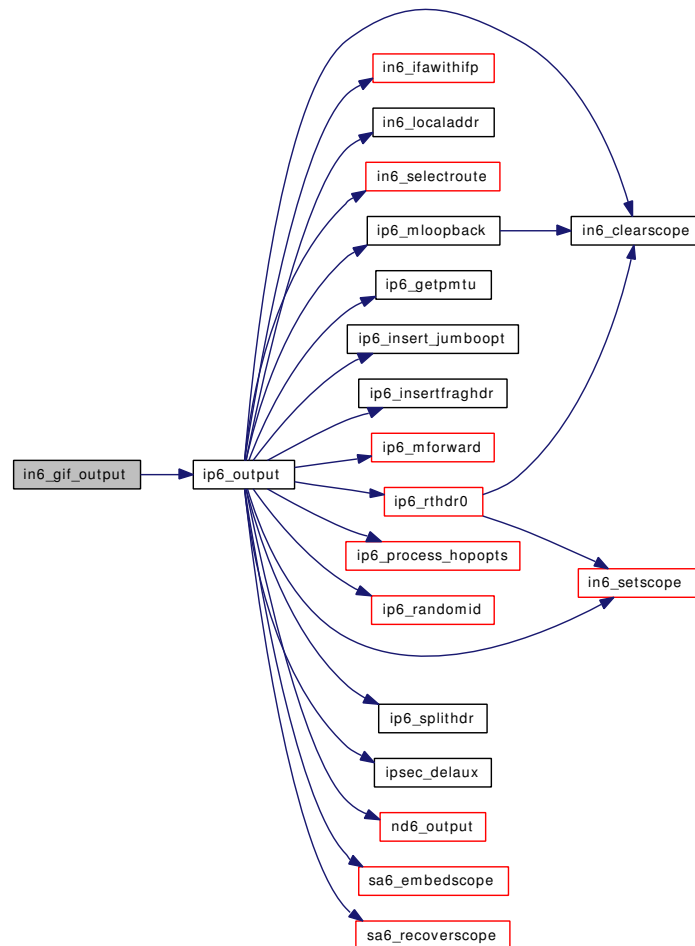


7.25.1.7 int in6_gif_output (struct ifnet * ifp, int family, struct mbuf * m)

Definition at line 84 of file in6_gif.c.

References IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_UNSPECIFIED, ip6_gif_hlim, ip6_output(), IPV6_MINMTU, sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_family.

Here is the call graph for this function:



7.25.2 Variable Documentation

7.25.2.1 struct ip6protosw in6_gif_protosw

Initial value:

```

{ SOCK_RAW,      &inet6domain,  0,      PR_ATOMIC|PR_ADDR,
  in6_gif_input, rip6_output,   0,      rip6_ctloutput,
  0,
  0,             0,             0,      0,
  &rip6_usrreqs
}

```

Definition at line 75 of file in6_gif.c.

Referenced by `in6_gif_attach()`.

7.25.2.2 struct domain `inet6domain`

Definition at line 369 of file `in6_proto.c`.

7.26 /usr/src/sys/netinet6/in6_gif.h File Reference

Defines

- #define [GIF_HLIM](#) 30

Functions

- int in6_gif_input __P ((struct mbuf **, int *, int))
- int in6_gif_output __P ((struct ifnet *, int, struct mbuf *))
- int gif_encapcheck6 __P ((const struct mbuf *, int, int, void *))
- int in6_gif_attach __P ((struct gif_softc *))

7.26.1 Define Documentation

7.26.1.1 #define GIF_HLIM 30

Definition at line 36 of file in6_gif.h.

7.26.2 Function Documentation

7.26.2.1 int in6_gif_detach __P ((struct gif_softc *))

7.26.2.2 int gif_encapcheck6 __P ((const struct mbuf *, int, int, void *))

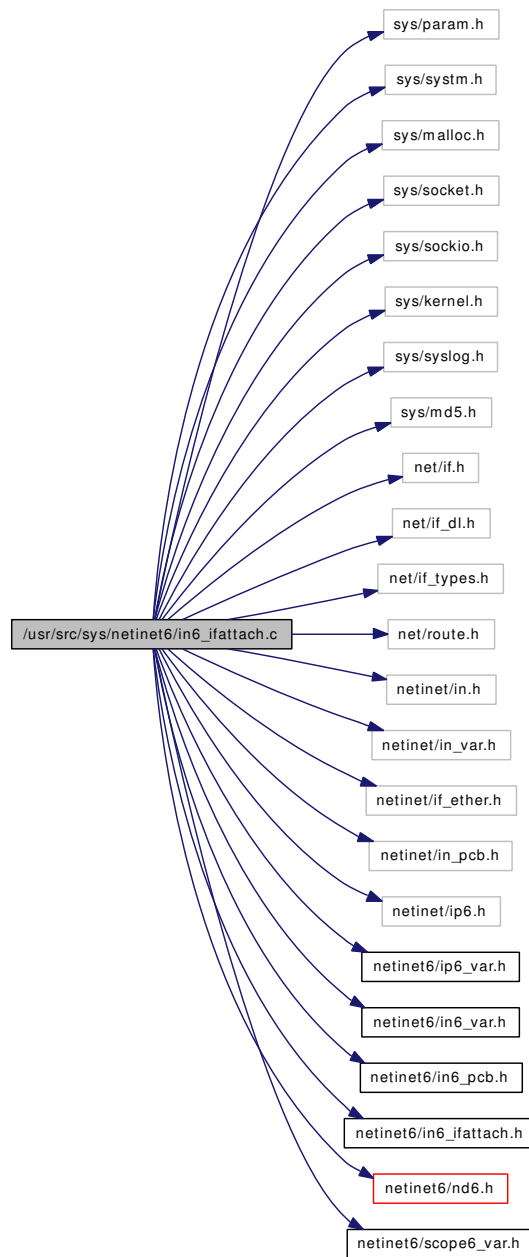
7.26.2.3 int in6_gif_output __P ((struct ifnet *, int, struct mbuf *))

7.26.2.4 int in6_gif_input __P ((struct mbuf **, int *, int))

7.27 /usr/src/sys/netinet6/in6_ifattach.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/socket.h>
#include <sys/sockio.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <sys/md5.h>
#include <net/if.h>
#include <net/if_dl.h>
#include <net/if_types.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/if_ether.h>
#include <netinet/in_pcb.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/in6_var.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/in6_ifattach.h>
#include <netinet6/nd6.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for in6_ifattach.c:



Defines

- #define [EUI64_GBIF](#) 0x01
- #define [EUI64_UBIF](#) 0x02
- #define [EUI64_TO_IFID](#)(in6) do { (in6) → s6_addr[8] ^= EUI64_UBIF; } while (0)
- #define [EUI64_GROUP](#)(in6) ((in6) → s6_addr[8] & EUI64_GBIF)
- #define [EUI64_INDIVIDUAL](#)(in6) (!EUI64_GROUP(in6))
- #define [EUI64_LOCAL](#)(in6) ((in6) → s6_addr[8] & EUI64_UBIF)
- #define [EUI64_UNIVERSAL](#)(in6) (!EUI64_LOCAL(in6))
- #define [IFID_LOCAL](#)(in6) (!EUI64_LOCAL(in6))
- #define [IFID_UNIVERSAL](#)(in6) (!EUI64_UNIVERSAL(in6))

Functions

- static int `get_rand_ifid` `__P` ((struct ifnet *, struct `in6_addr` *)
- static int `generate_tmp_ifid` `__P` ((u_int8_t *, const u_int8_t *, u_int8_t *)
- static int `get_ifid` `__P` ((struct ifnet *, struct ifnet *, struct `in6_addr` *)
- static int `in6_ifattach_linklocal` `__P` ((struct ifnet *, struct ifnet *)
- static int `in6_ifattach_loopback` `__P` ((struct ifnet *)
- static int `get_rand_ifid` (struct ifnet *ifp, struct `in6_addr` *in6)
- static int `generate_tmp_ifid` (u_int8_t *seed0, const u_int8_t *seed1, u_int8_t *ret)
- int `in6_get_hw_ifid` (struct ifnet *ifp, struct `in6_addr` *in6)
- static int `get_ifid` (struct ifnet *ifp0, struct ifnet *altifp, struct `in6_addr` *in6)
- static int `in6_ifattach_linklocal` (struct ifnet *ifp, struct ifnet *altifp)
- static int `in6_ifattach_loopback` (struct ifnet *ifp)
- int `in6_nigroup` (struct ifnet *ifp, const char *name, int namelen, struct `in6_addr` *in6)
- void `in6_ifattach` (struct ifnet *ifp, struct ifnet *altifp)
- void `in6_ifdetach` (struct ifnet *ifp)
- int `in6_get_tmpifid` (struct ifnet *ifp, u_int8_t *retbuf, const u_int8_t *baseid, int generate)
- void `in6_tmpaddrtimer` (void *ignored_arg)

Variables

- unsigned long `in6_maxmtu` = 0
- int `ip6_auto_linklocal` = 1
- callout `in6_tmpaddrtimer_ch`
- inpcbinfo `udbinfo`
- inpcbinfo `ripcbinfo`

7.27.1 Define Documentation

7.27.1.1 #define EUI64_GBIT 0x01

Definition at line 80 of file `in6_ifattach.c`.

Referenced by `get_rand_ifid()`, and `in6_get_hw_ifid()`.

7.27.1.2 #define EUI64_GROUP(in6) ((in6) → s6_addr[8] & EUI64_GBIT)

Definition at line 83 of file `in6_ifattach.c`.

Referenced by `in6_get_hw_ifid()`.

7.27.1.3 #define EUI64_INDIVIDUAL(in6) (!EUI64_GROUP(in6))

Definition at line 84 of file `in6_ifattach.c`.

7.27.1.4 #define EUI64_LOCAL(in6) ((in6) → s6_addr[8] & EUI64_UBIT)

Definition at line 85 of file `in6_ifattach.c`.

7.27.1.5 #define EUI64_TO_IFID(in6) do {(in6) → s6_addr[8] ^= EUI64_UBIT; } while (0)

Definition at line 82 of file in6_ifattach.c.

Referenced by get_rand_ifid(), and in6_get_hw_ifid().

7.27.1.6 #define EUI64_UBIT 0x02

Definition at line 81 of file in6_ifattach.c.

Referenced by generate_tmp_ifid(), get_rand_ifid(), and in6_get_hw_ifid().

7.27.1.7 #define EUI64_UNIVERSAL(in6) (!EUI64_LOCAL(in6))

Definition at line 86 of file in6_ifattach.c.

7.27.1.8 #define IFID_LOCAL(in6) (!EUI64_LOCAL(in6))

Definition at line 88 of file in6_ifattach.c.

7.27.1.9 #define IFID_UNIVERSAL(in6) (!EUI64_UNIVERSAL(in6))

Definition at line 89 of file in6_ifattach.c.

Referenced by get_ifid().

7.27.2 Function Documentation

7.27.2.1 static int in6_ifattach_loopback __P ((struct ifnet *)) [static]

7.27.2.2 static int in6_ifattach_linklocal __P ((struct ifnet *, struct ifnet *)) [static]

7.27.2.3 static int get_ifid __P ((struct ifnet *, struct ifnet *, struct in6_addr *)) [static]

7.27.2.4 static int generate_tmp_ifid __P ((u_int8_t *, const u_int8_t *, u_int8_t *)) [static]

7.27.2.5 static int get_rand_ifid __P ((struct ifnet *, struct in6_addr *)) [static]

7.27.2.6 static int generate_tmp_ifid (u_int8_t * seed0, const u_int8_t * seed1, u_int8_t * ret) [static]

Definition at line 133 of file in6_ifattach.c.

References EUI64_UBIT, and nd6log.

Referenced by in6_get_tmpifid(), and in6_tmpaddrtimer().

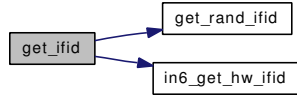
7.27.2.7 static int get_ifid (struct ifnet * ifp0, struct ifnet * altifp, struct in6_addr * in6) [static]

Definition at line 351 of file in6_ifattach.c.

References `get_rand_ifid()`, `IFID_UNIVERSAL`, `in6_get_hw_ifid()`, and `nd6log`.

Referenced by `in6_ifattach_linklocal()`.

Here is the call graph for this function:



7.27.2.8 `static int get_rand_ifid (struct ifnet * ifp, struct in6_addr * in6)` [static]

Definition at line 99 of file `in6_ifattach.c`.

References `EUI64_GBIT`, `EUI64_TO_IFID`, `EUI64_UBIT`, and `hostnamelen`.

Referenced by `get_ifid()`.

7.27.2.9 `int in6_get_hw_ifid (struct ifnet * ifp, struct in6_addr * in6)`

Definition at line 218 of file `in6_ifattach.c`.

References `EUI64_GBIT`, `EUI64_GROUP`, `EUI64_TO_IFID`, and `EUI64_UBIT`.

Referenced by `get_ifid()`, and `nd6_dad_duplicated()`.

7.27.2.10 `int in6_get_tmpifid (struct ifnet * ifp, u_int8_t * retbuf, const u_int8_t * baseid, int generate)`

Definition at line 838 of file `in6_ifattach.c`.

References `generate_tmp_ifid()`, `ND_IFINFO`, `nd_ifinfo::randomid`, `nd_ifinfo::randomseed0`, and `nd_ifinfo::randomseed1`.

Referenced by `in6_tmpifadd()`.

Here is the call graph for this function:



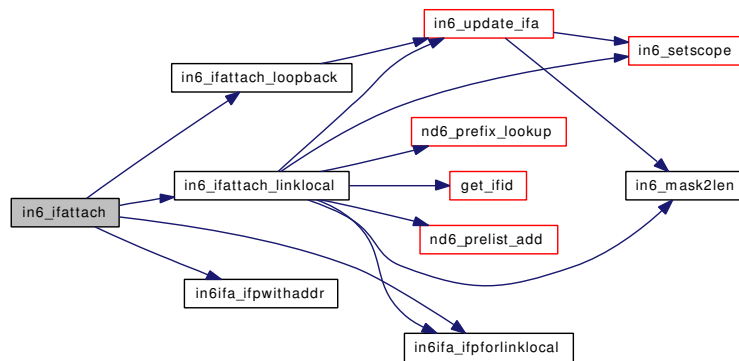
7.27.2.11 `void in6_ifattach (struct ifnet * ifp, struct ifnet * altifp)`

Definition at line 642 of file `in6_ifattach.c`.

References `in6_ifattach_linklocal()`, `in6_ifattach_loopback()`, `in6_maxmtu`, `in6addr_loopback`, `in6ifa_ifpforlinklocal()`, `in6ifa_ifpwithaddr()`, `ip6_auto_linklocal`, and `nd6log`.

Referenced by `in6_if_up()`.

Here is the call graph for this function:



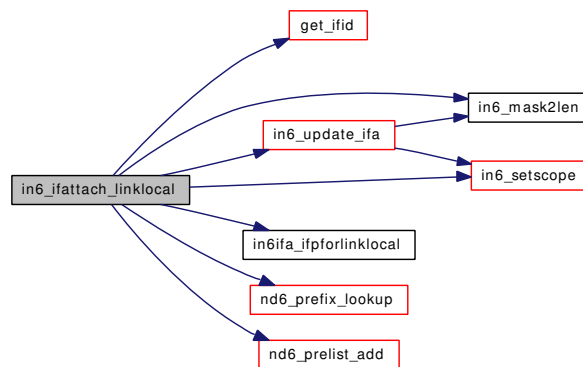
7.27.2.12 static int in6_ifattach_linklocal(struct ifnet * ifp, struct ifnet * altifp) [static]

Definition at line 414 of file in6_ifattach.c.

References `get_ifid()`, `IN6_IFAUPDATE_DADDELAY`, `in6_mask2len()`, `in6_setscope()`, `in6_update_ifa()`, `in6ifa_ifpforlinklocal()`, `in6mask64`, `ND6_INFINITE_LIFETIME`, `nd6_prefix_lookup()`, `nd6_prelist_add()`, and `nd6log`.

Referenced by `in6_ifattach()`.

Here is the call graph for this function:



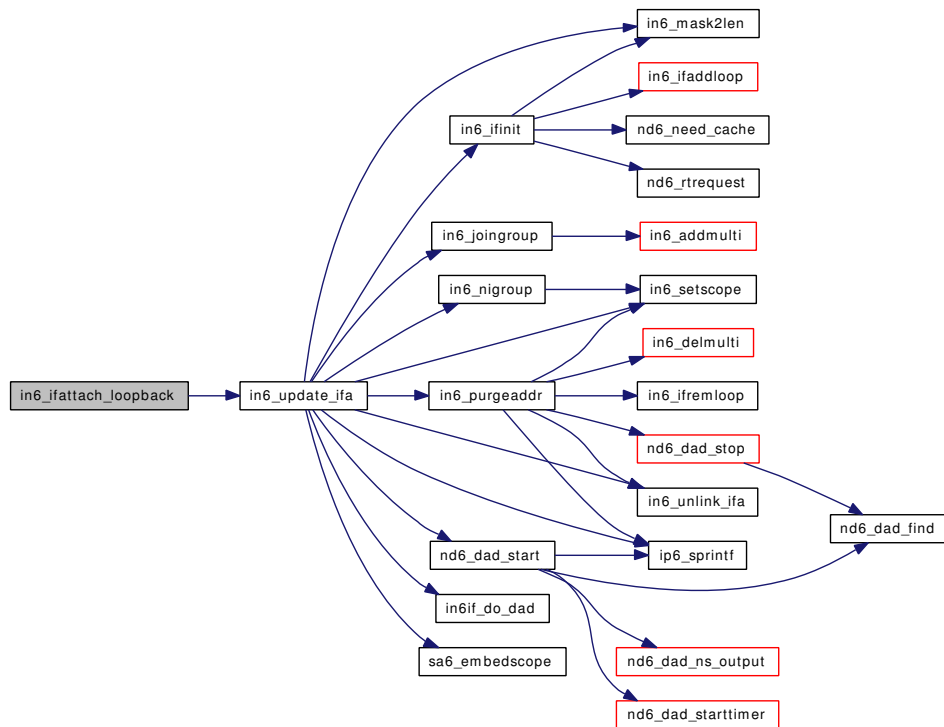
7.27.2.13 static int in6_ifattach_loopback(struct ifnet * ifp) [static]

Definition at line 529 of file in6_ifattach.c.

References `IN6_IFF_NODAD`, `IN6_IFF_NOPFX`, `in6_update_ifa()`, `in6addr_loopback`, `in6mask128`, `ND6_INFINITE_LIFETIME`, and `nd6log`.

Referenced by `in6_ifattach()`.

Here is the call graph for this function:



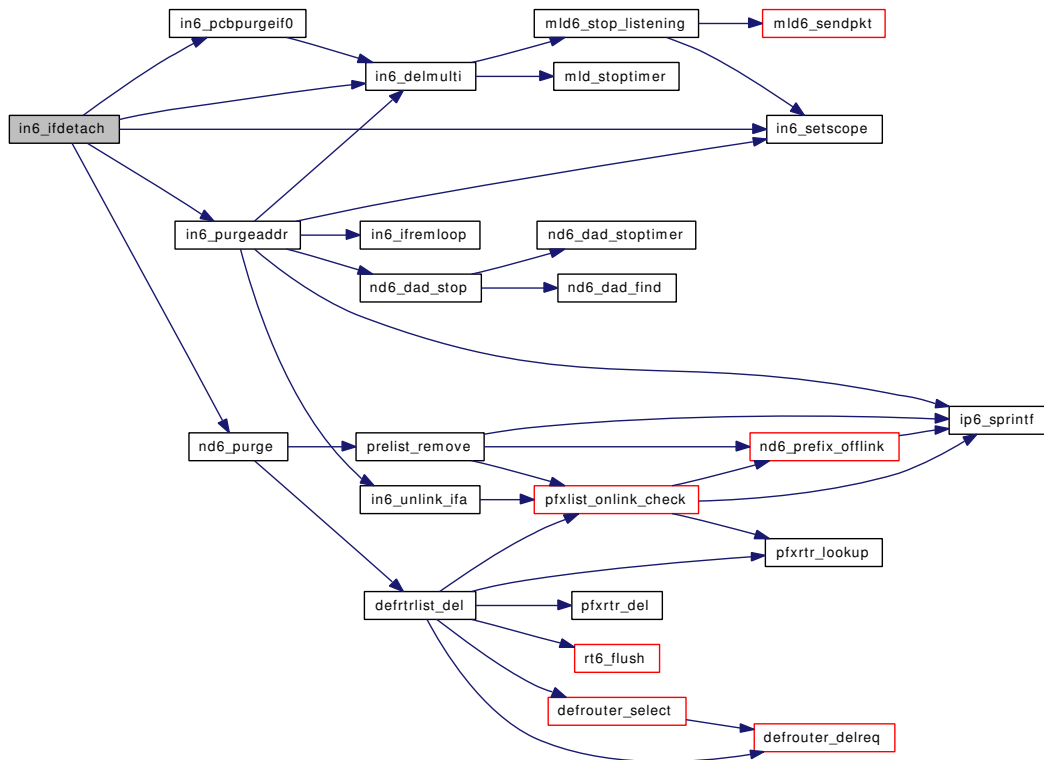
7.27.2.14 void in6_ifdetach (struct ifnet * ifp)

Definition at line 726 of file in6_ifattach.c.

References in6_ifaddr::ia_addr, in6_ifaddr::ia_ifa, in6_ifaddr::ia_next, in6_ifaddr::ia_prefixmask, in6_delmulti(), IN6_IS_ADDR_LINKLOCAL, in6_multihead, in6_pcbpurgeif0(), in6_purgeaddr(), in6_setscope(), in6addr_linklocal_allnodes, in6_multi::in6m_ifp, nd6_purge(), nd6log, ripcbinf, sato6sin6, sin6, and udbinf.

Referenced by in6_purgeif().

Here is the call graph for this function:



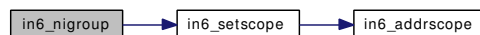
7.27.2.15 int in6_nigroup (struct ifnet * ifp, const char * name, int namelen, struct in6_addr * in6)

Definition at line 590 of file in6_ifattach.c.

References in6_setscope(), and IPV6_ADDR_INT16_MLL.

Referenced by in6_update_ifa().

Here is the call graph for this function:



7.27.2.16 void in6_tmpaddrtimer (void * ignored_arg)

Definition at line 866 of file in6_ifattach.c.

References generate_tmp_ifid(), in6_tmpaddrtimer_ch, ip6_desync_factor, ip6_temp_preferred_lifetime, ip6_temp_regen_advance, ND_IFINFO, nd_ifinfo::randomid, nd_ifinfo::randomseed0, and nd_ifinfo::randomseed1.

Referenced by ip6_init2().

Here is the call graph for this function:



7.27.3 Variable Documentation

7.27.3.1 unsigned long [in6_maxmtu](#) = 0

Definition at line 61 of file `in6_ifattach.c`.

Referenced by `in6_ifattach()`, `in6_setmaxmtu()`, and `nd6_setmtu0()`.

7.27.3.2 struct callout [in6_tmpaddrtimer_ch](#)

Definition at line 69 of file `in6_ifattach.c`.

Referenced by `in6_tmpaddrtimer()`, and `ip6_init2()`.

7.27.3.3 int [ip6_auto_linklocal](#) = 1

Definition at line 66 of file `in6_ifattach.c`.

Referenced by `in6_ifattach()`.

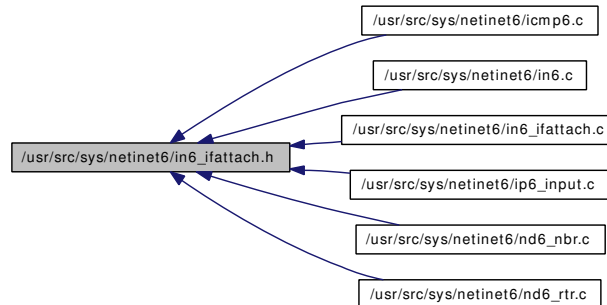
7.27.3.4 struct inpcbinfo [ripcbinfo](#)

7.27.3.5 struct inpcbinfo [udbinfo](#)

Referenced by `in6_ifdetach()`, `udp6_abort()`, `udp6_attach()`, `udp6_bind()`, `udp6_close()`, `udp6_connect()`, `udp6_ctlinput()`, `udp6_detach()`, `udp6_disconnect()`, `udp6_getcred()`, `udp6_input()`, and `udp6_send()`.

7.28 /usr/src/sys/netinet6/in6_ifattach.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void in6_ifattach __P ((struct ifnet *, struct ifnet *))
- void in6_ifdetach __P ((struct ifnet *))
- int in6_get_tmpifid __P ((struct ifnet *, u_int8_t *, const u_int8_t *, int))
- void in6_tmpaddrtimer __P ((void *))
- int in6_get_hw_ifid __P ((struct ifnet *, struct in6_addr *))
- int in6_nigroup __P ((struct ifnet *, const char *, int, struct in6_addr *))

7.28.1 Function Documentation

7.28.1.1 int in6_nigroup __P ((struct ifnet *, const char *, int, struct in6_addr *))

7.28.1.2 int in6_get_hw_ifid __P ((struct ifnet *, struct in6_addr *))

7.28.1.3 static void nd6_llinfo_timer __P ((void *))

7.28.1.4 int in6_get_tmpifid __P ((struct ifnet *, u_int8_t *, const u_int8_t *, int))

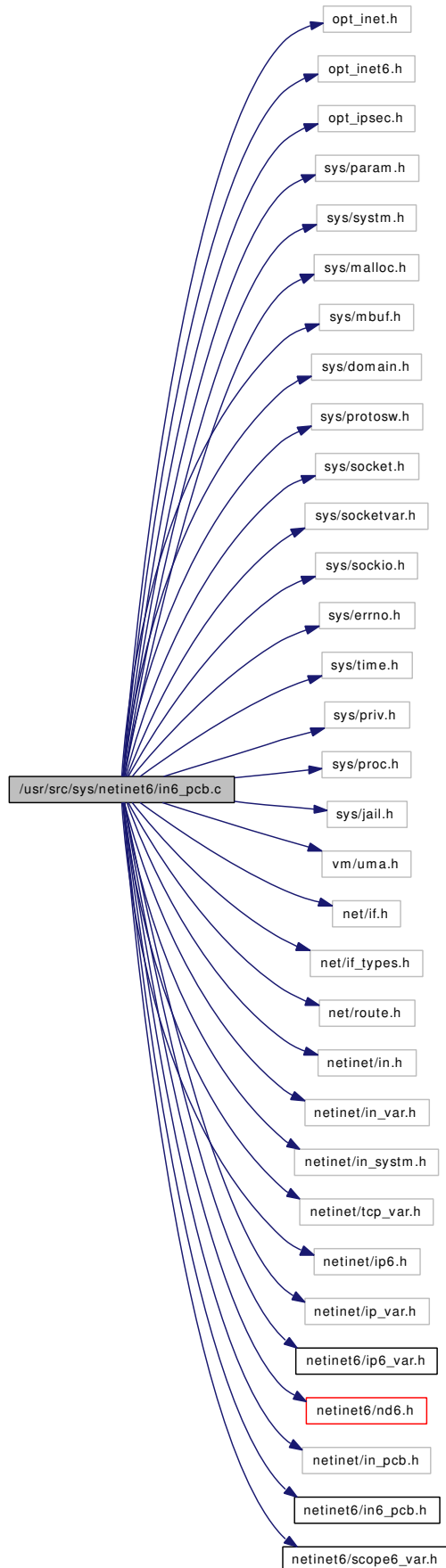
7.28.1.5 void in6_ifdetach __P ((struct ifnet *))

7.28.1.6 void in6_ifattach __P ((struct ifnet *, struct ifnet *))

7.29 /usr/src/sys/netinet6/in6_pcb.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sockio.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/priv.h>
#include <sys/proc.h>
#include <sys/jail.h>
#include <vm/uma.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/in_system.h>
#include <netinet/tcp_var.h>
#include <netinet/ip6.h>
#include <netinet/ip_var.h>
#include <netinet6/ip6_var.h>
#include <netinet6/nd6.h>
#include <netinet/in_pcb.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for in6_pcb.c:



Functions

- int [in6_pcbbind](#) (struct inpcb *inp, struct sockaddr *nam, struct ucred *cred)
- int [in6_pcbldaddr](#) (struct inpcb *inp, struct sockaddr *nam, struct [in6_addr](#) **plocal_addr6)
- int [in6_pcbconnect](#) (struct inpcb *inp, struct sockaddr *nam, struct ucred *cred)
- void [in6_pcbdisconnect](#) (struct inpcb *inp)
- void [in6_pcbdetach](#) (struct inpcb *inp)
- void [in6_pcbfree](#) (struct inpcb *inp)
- sockaddr * [in6_sockaddr](#) (in_port_t port, struct [in6_addr](#) *addr_p)
- sockaddr * [in6_v4mapsin6_sockaddr](#) (in_port_t port, struct in_addr *addr_p)
- int [in6_setsockaddr](#) (struct socket *so, struct sockaddr **nam)
- int [in6_setpeeraddr](#) (struct socket *so, struct sockaddr **nam)
- int [in6_mapped_sockaddr](#) (struct socket *so, struct sockaddr **nam)
- int [in6_mapped_peeraddr](#) (struct socket *so, struct sockaddr **nam)
- void [in6_pcbnotify](#) (struct inpcbinfo *pcbinfo, struct sockaddr *dst, u_int fport_arg, const struct sockaddr *src, u_int lport_arg, int cmd, void *cmdarg, struct inpcb **notify)
- inpcb * [in6_pcblookup_local](#) (struct inpcbinfo *pcbinfo, struct [in6_addr](#) *laddr, u_int lport_arg, int wild_okay)
- void [in6_pcbpurgeif0](#) (struct inpcbinfo *pcbinfo, struct ifnet *ifp)
- void [in6_losing](#) (struct inpcb *in6p)
- inpcb * [in6_rtchange](#) (struct inpcb *inp, int errno)
- inpcb * [in6_pcblookup_hash](#) (struct inpcbinfo *pcbinfo, struct [in6_addr](#) *faddr, u_int fport_arg, struct [in6_addr](#) *laddr, u_int lport_arg, int wildcard, struct ifnet *ifp)
- void [init_sin6](#) (struct [sockaddr_in6](#) *sin6, struct mbuf *m)

Variables

- [in6_addr zeroin6_addr](#)

7.29.1 Function Documentation

7.29.1.1 void [in6_losing](#) (struct inpcb * *in6p*)

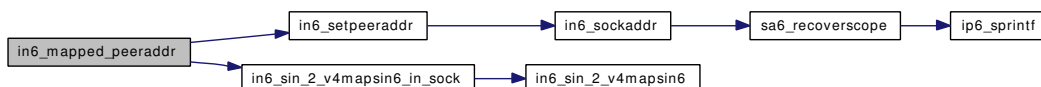
Definition at line 858 of file [in6_pcb.c](#).

7.29.1.2 int [in6_mapped_peeraddr](#) (struct socket * *so*, struct sockaddr ** *nam*)

Definition at line 580 of file [in6_pcb.c](#).

References [in6_setpeeraddr\(\)](#), and [in6_sin_2_v4mapsin6_in_sock\(\)](#).

Here is the call graph for this function:

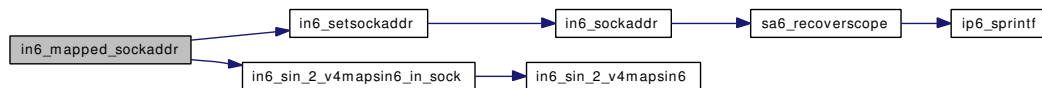


7.29.1.3 int in6_mapped_sockaddr (struct socket * so, struct sockaddr ** nam)

Definition at line 559 of file in6_pcb.c.

References in6_setsockaddr(), and in6_sin_2_v4mapsin6_in_sock().

Here is the call graph for this function:



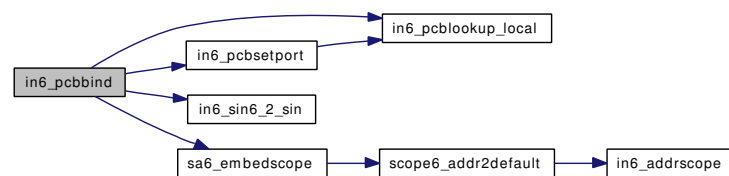
7.29.1.4 int in6_pcbbind (struct inpcb * inp, struct sockaddr * nam, struct ucred * cred)

Definition at line 123 of file in6_pcb.c.

References IN6_IFF_ANYCAST, IN6_IFF_DETACHED, IN6_IFF_NOTREADY, IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, in6_pcblookup_local(), in6_pcbsetport(), in6_sin6_2_sin(), in6addr_any, ip6_use_defzone, sa6_embedscope(), sin6, and sockaddr_in6::sin6_port.

Referenced by in6_pcbconnect(), and udp6_bind().

Here is the call graph for this function:



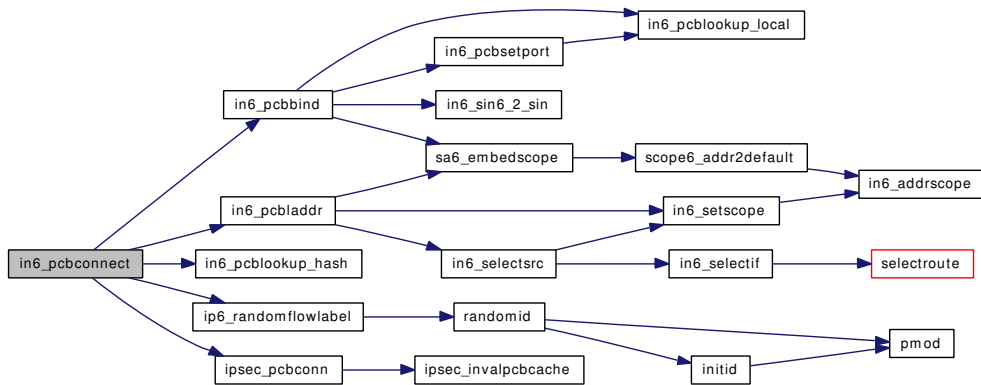
7.29.1.5 int in6_pcbconnect (struct inpcb * inp, struct sockaddr * nam, struct ucred * cred)

Definition at line 364 of file in6_pcb.c.

References IN6_IS_ADDR_UNSPECIFIED, in6_pcbbind(), in6_pcblookup_hash(), ip6_randomflowlabel(), ipsec_pcbconn(), sin6, sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_port.

Referenced by udp6_connect().

Here is the call graph for this function:



7.29.1.6 void in6_pcbdetach (struct inpcb * inp)

Definition at line 433 of file in6_pcb.c.

Referenced by rip6_detach(), and udp6_detach().

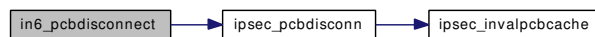
7.29.1.7 void in6_pcbdisconnect (struct inpcb * inp)

Definition at line 415 of file in6_pcb.c.

References ipsec_pcbdisconn().

Referenced by udp6_abort(), udp6_close(), and udp6_disconnect().

Here is the call graph for this function:



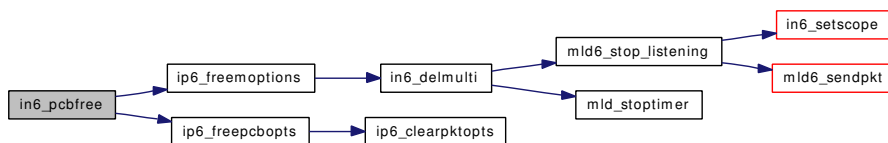
7.29.1.8 void in6_pcbfree (struct inpcb * inp)

Definition at line 442 of file in6_pcb.c.

References ip6_freemoptions(), and ip6_freepcbopts().

Referenced by rip6_detach(), and udp6_detach().

Here is the call graph for this function:



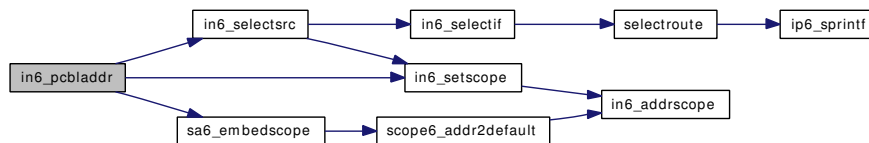
7.29.1.9 int in6_pcbaddr (struct inpcb * *inp*, struct sockaddr * *nam*, struct in6_addr ** *local_addr6*)

Definition at line 295 of file in6_pcb.c.

References IN6_IS_ADDR_UNSPECIFIED, in6_selectsrc(), in6_setscope(), in6addr_loopback, ip6_use_defzone, sa6_embedscope(), sin6, sockaddr_in6::sin6_addr, sockaddr_in6::sin6_family, sockaddr_in6::sin6_port, and sockaddr_in6::sin6_scope_id.

Referenced by in6_pcbconnect().

Here is the call graph for this function:



7.29.1.10 struct inpcb* in6_pcblookup_hash (struct inpcbinfo * *pcbinfo*, struct in6_addr * *faddr*, u_int *fport_arg*, struct in6_addr * *laddr*, u_int *lport_arg*, int *wildcard*, struct ifnet * *ifp*)

Definition at line 886 of file in6_pcb.c.

References faithprefix_p, IN6_ARE_ADDR_EQUAL, and IN6_IS_ADDR_UNSPECIFIED.

Referenced by in6_pcbconnect(), udp6_getcred(), and udp6_input().

7.29.1.11 struct inpcb* in6_pcblookup_local (struct inpcbinfo * *pcbinfo*, struct in6_addr * *laddr*, u_int *lport_arg*, int *wild_okay*)

Definition at line 721 of file in6_pcb.c.

References IN6_ARE_ADDR_EQUAL, and IN6_IS_ADDR_UNSPECIFIED.

Referenced by in6_pcbbind(), and in6_pcbsetport().

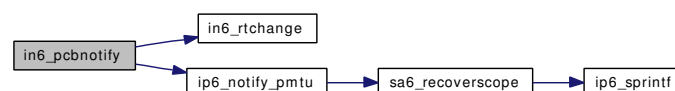
7.29.1.12 void in6_pcbnotify (struct inpcbinfo * *pcbinfo*, struct sockaddr * *dst*, u_int *fport_arg*, const struct sockaddr * *src*, u_int *lport_arg*, int *cmd*, void * *cmdarg*, struct inpcb ** *notify*)

Definition at line 609 of file in6_pcb.c.

References IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_UNSPECIFIED, in6_rtchange(), inet6ctlerrmap, ip6_notify_pmtu(), sa6_any, sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_flowinfo.

Referenced by rip6_ctlinput(), and udp6_ctlinput().

Here is the call graph for this function:



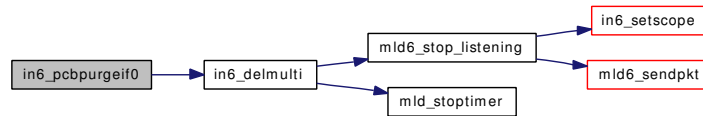
7.29.1.13 void in6_pcbpurgeif0 (struct inpcbinfo * *pcbinfo*, struct ifnet * *ifp*)

Definition at line 809 of file in6_pcb.c.

References in6_multi_mship::i6mm_maddr, ip6_moptions::im6o_multicast_ifp, in6_delmulti(), in6_multi::in6m_ifp, and in6pcb.

Referenced by in6_ifdetach().

Here is the call graph for this function:



7.29.1.14 struct inpcb* in6_rtchange (struct inpcb * *inp*, int *errno*)

Definition at line 872 of file in6_pcb.c.

Referenced by in6_pcbnotify(), rip6_ctlinput(), sctp6_ctlinput(), and udp6_ctlinput().

7.29.1.15 int in6_setpeeraddr (struct socket * *so*, struct sockaddr ** *nam*)

Definition at line 538 of file in6_pcb.c.

References in6_sockaddr().

Referenced by in6_mapped_peeraddr().

Here is the call graph for this function:



7.29.1.16 int in6_setsockaddr (struct socket * *so*, struct sockaddr ** *nam*)

Definition at line 517 of file in6_pcb.c.

References in6_sockaddr().

Referenced by in6_mapped_sockaddr().

Here is the call graph for this function:



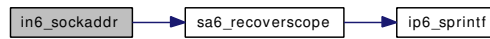
7.29.1.17 struct sockaddr* in6_sockaddr (in_port_t *port*, struct in6_addr * *addr_p*)

Definition at line 468 of file in6_pcb.c.

References `sa6_recoverscope()`, and `sin6`.

Referenced by `in6_setpeeraddr()`, and `in6_setsockaddr()`.

Here is the call graph for this function:



7.29.1.18 `struct sockaddr* in6_v4mapsin6_sockaddr(in_port_t port, struct in_addr * addr_p)`

Definition at line 486 of file `in6_pcb.c`.

References `in6_sin_2_v4mapsin6()`.

Here is the call graph for this function:



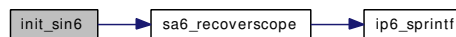
7.29.1.19 `void init_sin6(struct sockaddr_in6 * sin6, struct mbuf * m)`

Definition at line 953 of file `in6_pcb.c`.

References `sa6_recoverscope()`, and `sin6`.

Referenced by `rip6_input()`, and `udp6_input()`.

Here is the call graph for this function:



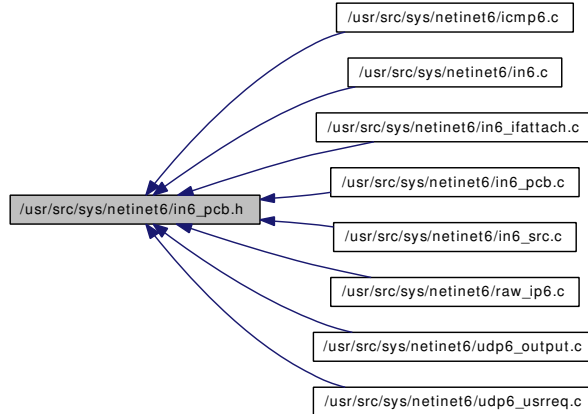
7.29.2 Variable Documentation

7.29.2.1 `struct in6_addr zeroin6_addr`

Definition at line 120 of file `in6_pcb.c`.

7.30 /usr/src/sys/netinet6/in6_pcb.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `satosin6(sa)` ((struct `sockaddr_in6` *)`(sa)`)
- #define `sin6tosa(sin6)` ((struct `sockaddr` *)`(sin6)`)
- #define `ifatoia6(ifa)` ((struct `in6_ifaddr` *)`(ifa)`)

Functions

- void `in6_pcbpurgeif0 __P` ((struct `inpcbinfo` *, struct `ifnet` *))
- void `in6_losing __P` ((struct `inpcb` *))
- int `in6_pcbbind __P` ((struct `inpcb` *, struct `sockaddr` *, struct `ucred` *))
- int `in6_pcbladdr __P` ((struct `inpcb` *, struct `sockaddr` *, struct `in6_addr` **))
- `inpcb *in6_pcblookup_local __P` ((struct `inpcbinfo` *, struct `in6_addr` *, u_int, int))
- `inpcb *in6_pcblookup_hash __P` ((struct `inpcbinfo` *, struct `in6_addr` *, u_int, struct `in6_addr` *, u_int, int, struct `ifnet` *))
- void `in6_pcbnotify __P` ((struct `inpcbinfo` *, struct `sockaddr` *, u_int, const struct `sockaddr` *, u_int, int, void *, struct `inpcb` *(*)`(struct inpcb *, int)`))
- `inpcb *in6_rtchange __P` ((struct `inpcb` *, int))
- `sockaddr *in6_sockaddr __P` ((in_port_t port, struct `in6_addr` *`addr_p`))
- `sockaddr *in6_v4mapsin6_sockaddr __P` ((in_port_t port, struct `in_addr` *`addr_p`))
- int `in6_setpeeraddr __P` ((struct `socket` *`so`, struct `sockaddr` **`nam`))
- int `in6_selectlim __P` ((struct `in6pcb` *, struct `ifnet` *))
- int `in6_pcbsetport __P` ((struct `in6_addr` *, struct `inpcb` *, struct `ucred` *))
- void `init_sin6 __P` ((struct `sockaddr_in6` *`sin6`, struct `mbuf` *`m`))

7.30.1 Define Documentation

7.30.1.1 #define `ifatoia6(ifa)` ((struct `in6_ifaddr` *)`(ifa)`)

Definition at line 71 of file `in6_pcb.h`.

7.30.1.2 #define satosin6(sa) ((struct sockaddr_in6 *) (sa))

Definition at line 69 of file in6_pcb.h.

7.30.1.3 #define sin6tosa(sin6) ((struct sockaddr *) (sin6))

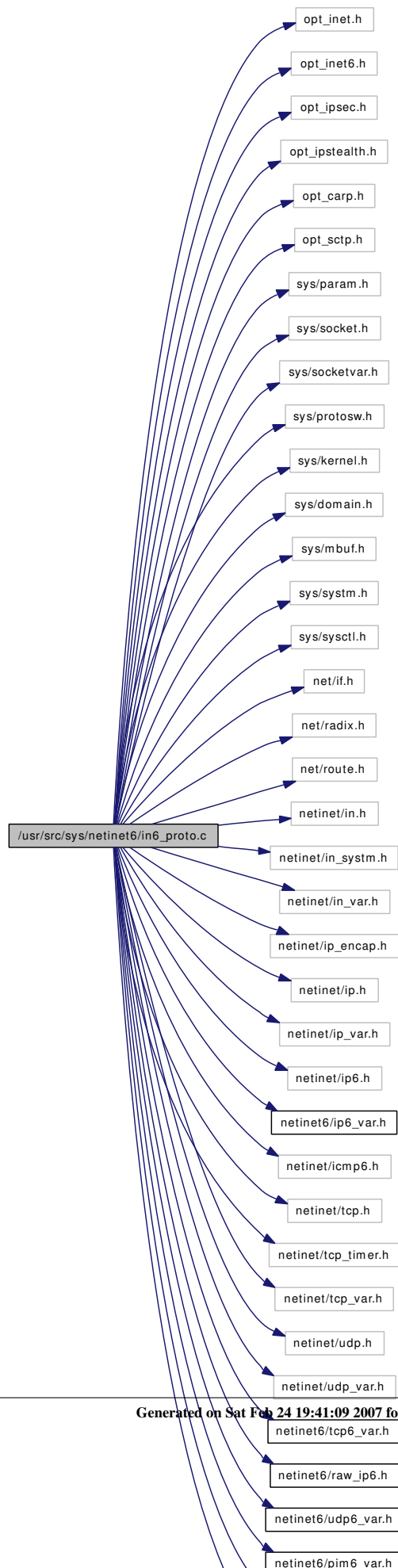
Definition at line 70 of file in6_pcb.h.

7.30.2 Function Documentation**7.30.2.1 void init_sin6 __P ((struct sockaddr_in6 *sin6, struct mbuf *m))****7.30.2.2 int in6_pcbsetport __P ((struct in6_addr *, struct inpcb *, struct ucred *))****7.30.2.3 int in6_selectthlim __P ((struct in6pcb *, struct ifnet *))****7.30.2.4 int in6_mapped_peeraddr __P ((struct socket *so, struct sockaddr **nam))****7.30.2.5 struct sockaddr* in6_v4mapsin6_sockaddr __P ((in_port_t port, struct in_addr *addr_p))****7.30.2.6 struct sockaddr* in6_sockaddr __P ((in_port_t port, struct in6_addr *addr_p))****7.30.2.7 struct inpcb* in6_rtchange __P ((struct inpcb *, int))****7.30.2.8 void in6_pcbnotify __P ((struct inpcbinfo *, struct sockaddr *, u_int, const struct sockaddr *, u_int, int, void *, struct inpcb *(*)(struct inpcb *, int)))****7.30.2.9 struct inpcb* in6_pcblookup_hash __P ((struct inpcbinfo *, struct in6_addr *, u_int, struct in6_addr *, u_int, int, struct ifnet *))****7.30.2.10 struct inpcb* in6_pcblookup_local __P ((struct inpcbinfo *, struct in6_addr *, u_int, int))****7.30.2.11 int in6_pcbldaddr __P ((struct inpcb *, struct sockaddr *, struct in6_addr **))****7.30.2.12 int in6_pcbconnect __P ((struct inpcb *, struct sockaddr *, struct ucred *))****7.30.2.13 void in6_pcbfree __P ((struct inpcb *))****7.30.2.14 void in6_pcbpurgeif0 __P ((struct inpcbinfo *, struct ifnet *))**

7.31 /usr/src/sys/netinet6/in6_proto.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include "opt_ipstealth.h"
#include "opt_carp.h"
#include "opt_sctp.h"
#include <sys/param.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/protosw.h>
#include <sys/kernel.h>
#include <sys/domain.h>
#include <sys/mbuf.h>
#include <sys/system.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <net/radix.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/in_var.h>
#include <netinet/ip_encap.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet/icmp6.h>
#include <netinet/tcp.h>
#include <netinet/tcp_timer.h>
#include <netinet/tcp_var.h>
#include <netinet/udp.h>
#include <netinet/udp_var.h>
#include <netinet6/tcp6_var.h>
#include <netinet6/raw_ip6.h>
#include <netinet6/udp6_var.h>
```

```
#include <netinet6/pim6_var.h>
#include <netinet6/nd6.h>
#include <netinet6/ip6protosw.h>
Include dependency graph for in6_proto.c:
```



Defines

- #define `PR_LISTEN` 0
- #define `PR_ABRTACPTDIS` 0
- #define `IPV6FORWARDING` 0
- #define `IPV6_SENDREDIRECTS` 1
- #define `RIPV6SNDQ` 8192
- #define `RIPV6RCVQ` 8192

Functions

- int `in6_inithead __P` ((void **, int))
- `DOMAIN_SET` (inet6)
- `SYSCTL_NODE` (_net, PF_INET6, inet6, CTLFLAG_RW, 0, "Internet6 Family")
- `SYSCTL_NODE` (_net_inet6, IPPROTO_IPV6, ip6, CTLFLAG_RW, 0, "IP6")
- `SYSCTL_NODE` (_net_inet6, IPPROTO_ICMPV6, icmp6, CTLFLAG_RW, 0, "ICMP6")
- `SYSCTL_NODE` (_net_inet6, IPPROTO_UDP, udp6, CTLFLAG_RW, 0, "UDP6")
- `SYSCTL_NODE` (_net_inet6, IPPROTO_TCP, tcp6, CTLFLAG_RW, 0, "TCP6")
- static int `sysctl_ip6_temppltime` (SYSCTL_HANDLER_ARGS)
- static int `sysctl_ip6_tempvlttime` (SYSCTL_HANDLER_ARGS)
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_FORWARDING, forwarding, CTLFLAG_RW, &ip6_forwarding, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_SENDREDIRECTS, redirect, CTLFLAG_RW, &ip6_sendredirects, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_DEFHLIM, hlim, CTLFLAG_RW, &ip6_defhlim, 0, "")
- `SYSCTL_STRUCT` (_net_inet6_ip6, IPV6CTL_STATS, stats, CTLFLAG_RD, &ip6stat, ip6stat, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_MAXFRAGPACKETS, maxfragpackets, CTLFLAG_RW, &ip6_maxfragpackets, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_ACCEPT_RTADV, accept_rtadv, CTLFLAG_RW, &ip6_accept_rtadv, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_KEEPPAITH, keepfaith, CTLFLAG_RW, &ip6_keepfaith, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_LOG_INTERVAL, log_interval, CTLFLAG_RW, &ip6_log_interval, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_HDRNESTLIMIT, hdrnestlimit, CTLFLAG_RW, &ip6_hdrnestlimit, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_DAD_COUNT, dad_count, CTLFLAG_RW, &ip6_dad_count, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_AUTO_FLOWLABEL, auto_flowlabel, CTLFLAG_RW, &ip6_auto_flowlabel, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_DEFMCASTHLIM, defmcasthlim, CTLFLAG_RW, &ip6_defmcasthlim, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_GIF_HLIM, gifhlim, CTLFLAG_RW, &ip6_gif_hlim, 0, "")
- `SYSCTL_STRING` (_net_inet6_ip6, IPV6CTL_KAME_VERSION, kame_version, CTLFLAG_RD, __KAME_VERSION, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_USE_DEPRECATED, use_deprecated, CTLFLAG_RW, &ip6_use_deprecated, 0, "")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_RR_PRUNE, rr_prune, CTLFLAG_RW, &ip6_rr_prune, 0, "")

- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_USETEMPADDR`, `use_tempaddr`, `CTLFLAG_RW,&ip6_use_tempaddr`, 0,"")
- `SYSCTL_OID` (`_net_inet6_ip6`, `IPV6CTL_TEMPPLTIME`, `temppltime`, `CTLTYPE_INT|CTLFLAG_RW,&ip6_temp_preferred_lifetime`, 0, `sysctl_ip6_temppltime`, "I", "")
- `SYSCTL_OID` (`_net_inet6_ip6`, `IPV6CTL_TEMPVLTIME`, `tempvltime`, `CTLTYPE_INT|CTLFLAG_RW,&ip6_temp_valid_lifetime`, 0, `sysctl_ip6_tempvltime`, "I", "")
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_V6ONLY`, `v6only`, `CTLFLAG_RW,&ip6_v6only`, 0,"")
- `TUNABLE_INT` ("net.inet6.ip6.auto_linklocal",&ip6_auto_linklocal)
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_AUTO_LINKLOCAL`, `auto_linklocal`, `CTLFLAG_RW,&ip6_auto_linklocal`, 0,"")
- `SYSCTL_STRUCT` (`_net_inet6_ip6`, `IPV6CTL_RIP6STATS`, `rip6stats`, `CTLFLAG_RD,&rip6stat`, `rip6stat`, "")
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_PREFER_TEMPADDR`, `prefer_tempaddr`, `CTLFLAG_RW,&ip6_prefer_tempaddr`, 0,"")
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_USE_DEFAULTZONE`, `use_defaultzone`, `CTLFLAG_RW,&ip6_use_defzone`, 0,"")
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_MAXFRAGS`, `maxfrags`, `CTLFLAG_RW,&ip6_maxfrags`, 0,"")
- `SYSCTL_INT` (`_net_inet6_ip6`, `IPV6CTL_MCAST_PMTU`, `mcast_pmtu`, `CTLFLAG_RW,&ip6_mcast_pmtu`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_REDIRACCEPT`, `rediraccept`, `CTLFLAG_RW,&icmp6_rediraccept`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_REDIRTIMEOUT`, `redirtimeout`, `CTLFLAG_RW,&icmp6_redirtimeout`, 0,"")
- `SYSCTL_STRUCT` (`_net_inet6_icmp6`, `ICMPV6CTL_STATS`, `stats`, `CTLFLAG_RD,&icmp6stat`, `icmp6stat`, "")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_PRUNE`, `nd6_prune`, `CTLFLAG_RW,&nd6_prune`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_DELAY`, `nd6_delay`, `CTLFLAG_RW,&nd6_delay`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_UMAXTRIES`, `nd6_umaxtries`, `CTLFLAG_RW,&nd6_umaxtries`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_MMAXTRIES`, `nd6_mmaxtries`, `CTLFLAG_RW,&nd6_mmaxtries`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_USELOOPBACK`, `nd6_useloopback`, `CTLFLAG_RW,&nd6_useloopback`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_NODEINFO`, `nodeinfo`, `CTLFLAG_RW,&icmp6_nodeinfo`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ERRPPSLIMIT`, `errppslimit`, `CTLFLAG_RW,&icmp6errppslim`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_MAXNUDHINT`, `nd6_maxnudhint`, `CTLFLAG_RW,&nd6_maxnudhint`, 0,"")
- `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_DEBUG`, `nd6_debug`, `CTLFLAG_RW,&nd6_debug`, 0,"")

Variables

- domain `inet6domain`
- static struct `pr_usrreqs` `nousrreqs`
- `ip6protosw` `inet6sw` []
- domain `inet6domain`

- int `ip6_forwarding` = IPV6FORWARDING
- int `ip6_sendredirects` = IPV6_SENDREDIRECTS
- int `ip6_defhlim` = IPV6_DEFHLIM
- int `ip6_defmcasthlim` = IPV6_DEFAULT_MULTICAST_HOPS
- int `ip6_accept_rtadv` = 0
- int `ip6_maxfragpackets`
- int `ip6_maxfrags`
- int `ip6_log_interval` = 5
- int `ip6_hdrnestlimit` = 50
- int `ip6_dad_count` = 1
- int `ip6_auto_flowlabel` = 1
- int `ip6_gif_hlim` = 0
- int `ip6_use_deprecated` = 1
- int `ip6_rr_prune` = 5
- int `ip6_mcast_pmtu` = 0
- int `ip6_v6only` = 1
- int `ip6_keepfaith` = 0
- time_t `ip6_log_time` = (time_t)0L
- int `pmtu_expire` = 60*10
- int `pmtu_probe` = 60*2
- u_long `rip6_sendspace` = RIPV6SNDQ
- u_long `rip6_recvspace` = RIPV6RCVQ
- int `icmp6_rediraccept` = 1
- int `icmp6_redirtimeout` = 10 * 60
- int `icmp6errpplim` = 100
- int `icmp6_nodeinfo` = 3
- int `udp6_sendspace` = 9216
- int `udp6_recvspace` = 40 * (1024 + sizeof(struct `sockaddr_in6`))

7.31.1 Define Documentation

7.31.1.1 #define IPV6_SENDREDIRECTS 1

Definition at line 396 of file `in6_proto.c`.

7.31.1.2 #define IPV6FORWARDING 0

Definition at line 391 of file `in6_proto.c`.

7.31.1.3 #define PR_ABRTACPTDIS 0

Definition at line 158 of file `in6_proto.c`.

7.31.1.4 #define PR_LISTEN 0

Definition at line 157 of file `in6_proto.c`.

7.31.1.5 #define RIPv6RCVQ 8192

Definition at line 437 of file in6_proto.c.

7.31.1.6 #define RIPv6SNDQ 8192

Definition at line 436 of file in6_proto.c.

7.31.2 Function Documentation

- 7.31.2.1 `int in6_inithead __P((void **, int))`
- 7.31.2.2 `DOMAIN_SET (inet6)`
- 7.31.2.3 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_DEBUG, nd6_debug, CTLFLAG_RW, & nd6_debug, 0, "")`
- 7.31.2.4 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_MAXNUDHINT, nd6_maxnudhint, CTLFLAG_RW, & nd6_maxnudhint, 0, "")`
- 7.31.2.5 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ERRPPSLIMIT, errppslimit, CTLFLAG_RW, & icmp6errppslim, 0, "")`
- 7.31.2.6 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_NODEINFO, nodeinfo, CTLFLAG_RW, & icmp6_nodeinfo, 0, "")`
- 7.31.2.7 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_USELOOPBACK, nd6_useloopback, CTLFLAG_RW, & nd6_useloopback, 0, "")`
- 7.31.2.8 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_MMAXTRIES, nd6_mmaxtries, CTLFLAG_RW, & nd6_mmaxtries, 0, "")`
- 7.31.2.9 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_UMAXTRIES, nd6_umaxtries, CTLFLAG_RW, & nd6_umaxtries, 0, "")`
- 7.31.2.10 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_DELAY, nd6_delay, CTLFLAG_RW, & nd6_delay, 0, "")`
- 7.31.2.11 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_ND6_PRUNE, nd6_prune, CTLFLAG_RW, & nd6_prune, 0, "")`
- 7.31.2.12 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_REDIRTIMEOUT, redirtimeout, CTLFLAG_RW, & icmp6_redirtimeout, 0, "")`
- 7.31.2.13 `SYSCTL_INT (_net_inet6_icmp6, ICMPV6CTL_REDIRACCEPT, rediraccept, CTLFLAG_RW, & icmp6_rediraccept, 0, "")`
- 7.31.2.14 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_MCAST_PMTU, mcast_pmtu, CTLFLAG_RW, & ip6_mcast_pmtu, 0, "")`
- 7.31.2.15 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_MAXFRAGS, maxfrags, CTLFLAG_RW, & ip6_maxfrags, 0, "")`
- 7.31.2.16 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_USE_DEFAULTZONE, use_defaultzone, CTLFLAG_RW, & ip6_use_defzone, 0, "")`
- 7.31.2.17 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_PREFER_TEMPADDR, prefer_tempaddr, CTLFLAG_RW, & ip6_prefer_tempaddr, 0, "")`
- 7.31.2.18 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_AUTO_LINKLOCAL, auto_linklocal, CTLFLAG_RW, & ip6_auto_linklocal, 0, "")`
- 7.31.2.19 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_V6ONLY, v6only, CTLFLAG_RW, & ip6_v6only, 0, "")`
- 7.31.2.20 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_USETEMPADDR, use_tempaddr, CTLFLAG_RW, & ip6_use_tempaddr, 0, "")`
- 7.31.2.21 `SYSCTL_INT (_net_inet6_ip6, IPV6CTL_RR_PRUNE, rr_prune, CTLFLAG_RW, &`

References `ip6_desync_factor`, `ip6_temp_preferred_lifetime`, and `ip6_temp_regen_advance`.

7.31.2.36 `static int sysctl_ip6_tempvltime (SYSCTL_HANDLER_ARGS)` [static]

Definition at line 492 of file `in6_proto.c`.

References `ip6_temp_preferred_lifetime`, and `ip6_temp_valid_lifetime`.

7.31.2.37 `SYSCTL_NODE (_net_inet6, IPPROTO_TCP, tcp6, CTLFLAG_RW, 0, "TCP6")`

7.31.2.38 `SYSCTL_NODE (_net_inet6, IPPROTO_UDP, udp6, CTLFLAG_RW, 0, "UDP6")`

7.31.2.39 `SYSCTL_NODE (_net_inet6, IPPROTO_ICMPV6, icmp6, CTLFLAG_RW, 0, "ICMP6")`

7.31.2.40 `SYSCTL_NODE (_net_inet6, IPPROTO_IPV6, ip6, CTLFLAG_RW, 0, "IP6")`

7.31.2.41 `SYSCTL_NODE (_net, PF_INET6, inet6, CTLFLAG_RW, 0, "Internet6 Family")`

7.31.2.42 `SYSCTL_OID (_net_inet6_ip6, IPV6CTL_TEMPVLTIME, tempvltime, CTLTYPE_INT|CTLFLAG_RW, &ip6_temp_valid_lifetime, 0, sysctl_ip6_tempvltime, "I", "")`

7.31.2.43 `SYSCTL_OID (_net_inet6_ip6, IPV6CTL_TEMPPLTIME, temppltime, CTLTYPE_INT|CTLFLAG_RW, &ip6_temp_preferred_lifetime, 0, sysctl_ip6_temppltime, "I", "")`

7.31.2.44 `SYSCTL_STRING (_net_inet6_ip6, IPV6CTL_KAME_VERSION, kame_version, CTLFLAG_RD, __KAME_VERSION, 0, "")`

7.31.2.45 `SYSCTL_STRUCT (_net_inet6_icmp6, ICMPV6CTL_STATS, stats, CTLFLAG_RD, &icmp6stat, icmp6stat, "")`

7.31.2.46 `SYSCTL_STRUCT (_net_inet6_ip6, IPV6CTL_RIP6STATS, rip6stats, CTLFLAG_RD, &rip6stat, rip6stat, "")`

7.31.2.47 `SYSCTL_STRUCT (_net_inet6_ip6, IPV6CTL_STATS, stats, CTLFLAG_RD, &ip6stat, ip6stat, "")`

7.31.2.48 `TUNABLE_INT ("net.inet6.ip6.auto_linklocal", &ip6_auto_linklocal)`

7.31.3 Variable Documentation

7.31.3.1 `int icmp6_nodeinfo = 3`

Definition at line 446 of file `in6_proto.c`.

Referenced by `icmp6_input()`, `ni6_addrs()`, `ni6_input()`, and `ni6_store_addrs()`.

7.31.3.2 `int icmp6_rediraccept = 1`

Definition at line 443 of file `in6_proto.c`.

Referenced by `icmp6_redirect_input()`.

7.31.3.3 `int icmp6_redirtimeout = 10 * 60`

Definition at line 444 of file `in6_proto.c`.

7.31.3.4 `int icmp6errppslim = 100`

Definition at line 445 of file `in6_proto.c`.

Referenced by `icmp6_ratelimit()`.

7.31.3.5 `struct domain inet6domain`

Initial value:

```
{
    .dom_family =          AF_INET6,
    .dom_name =           "internet6",
    .dom_protosw =        (struct protosw *)inet6sw,
    .dom_protoswNPROTOSW = (struct protosw *)
        &inet6sw[sizeof(inet6sw)/sizeof(inet6sw[0])],
    .dom_rtattach =       in6_inithead,
    .dom_rtoffset =       offsetof(struct sockaddr_in6, sin6_addr) << 3,
    .dom_maxrtkey =       sizeof(struct sockaddr_in6),
    .dom_ifattach =       in6_domifattach,
    .dom_ifdetach =       in6_domifdetach
}
```

Definition at line 369 of file `in6_proto.c`.

7.31.3.6 `struct domain inet6domain`

Definition at line 369 of file `in6_proto.c`.

7.31.3.7 `struct ip6protosw inet6sw[]`

Definition at line 160 of file `in6_proto.c`.

Referenced by `icmp6_notify_error()`, `ip6_init()`, and `ip6_input()`.

7.31.3.8 `int ip6_accept_rtadv = 0`

Definition at line 403 of file `in6_proto.c`.

Referenced by `defrouter_select()`, `defrtrlist_del()`, `nd6_cache_lladdr()`, `nd6_ra_input()`, and `nd6_rs_input()`.

7.31.3.9 `int ip6_auto_flowlabel = 1`

Definition at line 409 of file `in6_proto.c`.

7.31.3.10 int ip6_dad_count = 1

Definition at line 408 of file in6_proto.c.

Referenced by nd6_dad_start().

7.31.3.11 int ip6_defhlim = IPV6_DEFHLIM

Definition at line 401 of file in6_proto.c.

Referenced by icmp6_reflect(), and in6_selecthlim().

7.31.3.12 int ip6_defmcasthlim = IPV6_DEFAULT_MULTICAST_HOPS

Definition at line 402 of file in6_proto.c.

Referenced by ip6_getmoptions(), ip6_output(), and ip6_setmoptions().

7.31.3.13 int ip6_forwarding = IPV6FORWARDING

Definition at line 399 of file in6_proto.c.

Referenced by defrouter_select(), defrtrlist_del(), icmp6_redirect_input(), icmp6_redirect_output(), ip6_input(), nd6_cache_lladdr(), nd6_free(), nd6_is_new_addr_neighbor(), nd6_na_input(), nd6_ns_input(), and nd6_rs_input().

7.31.3.14 int ip6_gif_hlim = 0

Definition at line 410 of file in6_proto.c.

Referenced by in6_gif_output().

7.31.3.15 int ip6_hdrnestlimit = 50

Definition at line 407 of file in6_proto.c.

Referenced by ip6_input().

7.31.3.16 int ip6_keepfaith = 0

Definition at line 417 of file in6_proto.c.

Referenced by ip6_input().

7.31.3.17 int ip6_log_interval = 5

Definition at line 406 of file in6_proto.c.

Referenced by ip6_forward(), and ip6_mforward().

7.31.3.18 `time_t ip6_log_time = (time_t)0L`

Definition at line 418 of file in6_proto.c.

Referenced by ip6_forward(), and ip6_mforward().

7.31.3.19 `int ip6_maxfragpackets`

Definition at line 404 of file in6_proto.c.

Referenced by frag6_change(), frag6_init(), frag6_input(), and frag6_slowtimo().

7.31.3.20 `int ip6_maxfrags`

Definition at line 405 of file in6_proto.c.

Referenced by frag6_change(), frag6_init(), and frag6_input().

7.31.3.21 `int ip6_mcast_pmtu = 0`

Definition at line 414 of file in6_proto.c.

Referenced by phyint_send().

7.31.3.22 `int ip6_rr_prune = 5`

Definition at line 412 of file in6_proto.c.

7.31.3.23 `int ip6_sendredirects = IPV6_SENDREDIRECTS`

Definition at line 400 of file in6_proto.c.

Referenced by ip6_forward().

7.31.3.24 `int ip6_use_deprecated = 1`

Definition at line 411 of file in6_proto.c.

Referenced by in6_ifawithifp(), and in6_selectsrc().

7.31.3.25 `int ip6_v6only = 1`

Definition at line 415 of file in6_proto.c.

Referenced by sctp6_connect(), and sctp6_send().

7.31.3.26 `struct pr_usrreqs nousrreqs` `[static]`

Definition at line 155 of file in6_proto.c.

7.31.3.27 `int pmtu_expire = 60*10`

Definition at line 429 of file in6_proto.c.

7.31.3.28 `int pmtu_probe = 60*2`

Definition at line 430 of file in6_proto.c.

7.31.3.29 `u_long rip6_recvspace = RIPV6RCVQ`

Definition at line 440 of file in6_proto.c.

7.31.3.30 `u_long rip6_sendspace = RIPV6SNDQ`

Definition at line 439 of file in6_proto.c.

7.31.3.31 `int udp6_recvspace = 40 * (1024 + sizeof(struct sockaddr_in6))`

Definition at line 450 of file in6_proto.c.

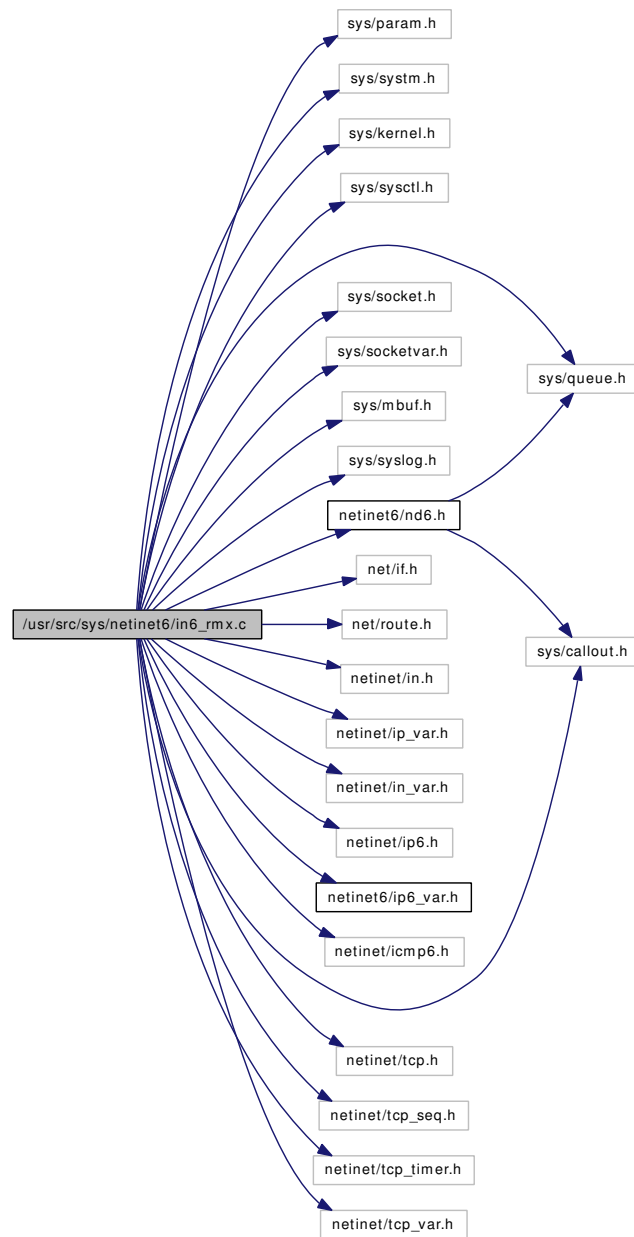
7.31.3.32 `int udp6_sendspace = 9216`

Definition at line 449 of file in6_proto.c.

7.32 /usr/src/sys/netinet6/in6_rmx.c File Reference

```
#include <sys/param.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <sys/sysctl.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/mbuf.h>
#include <sys/syslog.h>
#include <sys/callout.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/ip_var.h>
#include <netinet/in_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet/icmp6.h>
#include <netinet6/nd6.h>
#include <netinet/tcp.h>
#include <netinet/tcp_seq.h>
#include <netinet/tcp_timer.h>
#include <netinet/tcp_var.h>
```

Include dependency graph for in6_rmx.c:



Data Structures

- struct [rtqk_arg](#)
- struct [mtuex_arg](#)

Defines

- #define [RTPRF_OURS](#) RTF_PROTO3
- #define [RTQ_TIMEOUT](#) 60*10
- #define [MTUTIMO_DEFAULT](#) (60*1)

Functions

- int `in6_inithead` `__P` ((void **head, int off))
- static struct radix_node * `in6_addroute` (void *v_arg, void *n_arg, struct radix_node_head *head, struct radix_node *treenodes)
- static struct radix_node * `in6_matroute` (void *v_arg, struct radix_node_head *head)
- `SYSCTL_DECL` (_net_inet6_ip6)
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_RTEXPIRE, rtxpire, CTLFLAG_RW,&rtq_reallyold, 0,"")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_RTMINEXPIRE, rtminexpire, CTLFLAG_RW,&rtq_minreallyold, 0,"")
- `SYSCTL_INT` (_net_inet6_ip6, IPV6CTL_RTMAXCACHE, rtmaxcache, CTLFLAG_RW,&rtq_toomany, 0,"")
- static void `in6_clsroute` (struct radix_node *rn, struct radix_node_head *head)
- static int `in6_rtqkill` (struct radix_node *rn, void *rock)
- static void `in6_rtqtime` (void *rock)
- static int `in6_mtuexpire` (struct radix_node *rn, void *rock)
- static void `in6_mtutimo` (void *rock)
- int `in6_inithead` (void **head, int off)

Variables

- static int `rtq_reallyold` = 60*60
- static int `rtq_minreallyold` = 10
- static int `rtq_toomany` = 128
- static int `rtq_timeout` = RTQ_TIMEOUT
- static struct callout `rtq_timer`
- static struct callout `rtq_mtutimer`

7.32.1 Define Documentation

7.32.1.1 #define MTUTIMO_DEFAULT (60*1)

Definition at line 405 of file `in6_rmx.c`.

Referenced by `in6_mtutimo()`.

7.32.1.2 #define RTPRF_OURS RTF_PROTO3

Definition at line 106 of file `in6_rmx.c`.

Referenced by `in6_clsroute()`, `in6_matroute()`, and `in6_rtqkill()`.

7.32.1.3 #define RTQ_TIMEOUT 60*10

Definition at line 321 of file `in6_rmx.c`.

7.32.2 Function Documentation

7.32.2.1 `int in6_inithead __P((void **head, int off))`

7.32.2.2 `static struct radix_node* in6_addroute (void * v_arg, void * n_arg, struct radix_node_head * head, struct radix_node * treenodes) [static]`

Definition at line 112 of file in6_rmx.c.

References IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_MULTICAST, IN6_LINKMTU, satoSin6, sin6, and sockaddr_in6::sin6_addr.

Referenced by in6_inithead().

7.32.2.3 `static void in6_clsroute (struct radix_node * rn, struct radix_node_head * head) [static]`

Definition at line 240 of file in6_rmx.c.

References RTPRF_OURS.

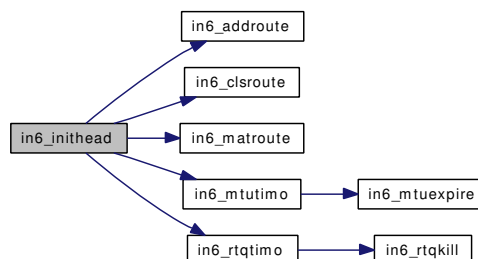
Referenced by in6_inithead().

7.32.2.4 `int in6_inithead (void ** head, int off)`

Definition at line 452 of file in6_rmx.c.

References in6_addroute(), in6_clsroute(), in6_matroute(), in6_mtutimo(), and in6_rtqtimeo().

Here is the call graph for this function:



7.32.2.5 `static struct radix_node* in6_matroute (void * v_arg, struct radix_node_head * head) [static]`

Definition at line 203 of file in6_rmx.c.

References RTPRF_OURS.

Referenced by in6_inithead().

7.32.2.6 `static int in6_mtuexpire (struct radix_node * rn, void * rock) [static]`

Definition at line 384 of file in6_rmx.c.

References `mtuex_arg::nextstop`.

Referenced by `in6_mtutimo()`.

7.32.2.7 `static void in6_mtutimo (void * rock) [static]`

Definition at line 408 of file `in6_rmx.c`.

References `in6_mtuexpire()`, `MTUTIMO_DEFAULT`, `mtuex_arg::nextstop`, and `mtuex_arg::rnh`.

Referenced by `in6_inithead()`.

Here is the call graph for this function:



7.32.2.8 `static int in6_rtqkill (struct radix_node * rn, void * rock) [static]`

Definition at line 284 of file `in6_rmx.c`.

References `rtqk_arg::draining`, `rtqk_arg::found`, `rtqk_arg::killed`, `rtqk_arg::nextstop`, `RTPRF_OURS`, and `rtqk_arg::updating`.

Referenced by `in6_rtqtimeo()`.

7.32.2.9 `static void in6_rtqtimeo (void * rock) [static]`

Definition at line 326 of file `in6_rmx.c`.

References `rtqk_arg::draining`, `rtqk_arg::found`, `in6_rtqkill()`, `rtqk_arg::killed`, `rtqk_arg::nextstop`, `rtqk_arg::rnh`, and `rtqk_arg::updating`.

Referenced by `in6_inithead()`.

Here is the call graph for this function:



7.32.2.10 **SYSCTL_DECL** (`_net_inet6_ip6`)

7.32.2.11 **SYSCTL_INT** (`_net_inet6_ip6`, `IPV6CTL_RTMAXCACHE`, `rtmaxcache`, `CTLFLAG_RW`, & `rtq_toomany`, 0, "")

7.32.2.12 **SYSCTL_INT** (`_net_inet6_ip6`, `IPV6CTL_RTMINEXPIRE`, `rtminexpire`, `CTLFLAG_RW`, & `rtq_minreallyold`, 0, "")

7.32.2.13 **SYSCTL_INT** (`_net_inet6_ip6`, `IPV6CTL_RTEXPIRE`, `rtexpire`, `CTLFLAG_RW`, & `rtq_reallyold`, 0, "")

7.32.3 Variable Documentation

7.32.3.1 **int** `rtq_minreallyold` = 10 [static]

Definition at line 224 of file `in6_rmx.c`.

7.32.3.2 **struct callout** `rtq_mtutimer` [static]

Definition at line 381 of file `in6_rmx.c`.

7.32.3.3 **int** `rtq_reallyold` = 60*60 [static]

Definition at line 219 of file `in6_rmx.c`.

7.32.3.4 **int** `rtq_timeout` = `RTQ_TIMEOUT` [static]

Definition at line 322 of file `in6_rmx.c`.

7.32.3.5 **struct callout** `rtq_timer` [static]

Definition at line 323 of file `in6_rmx.c`.

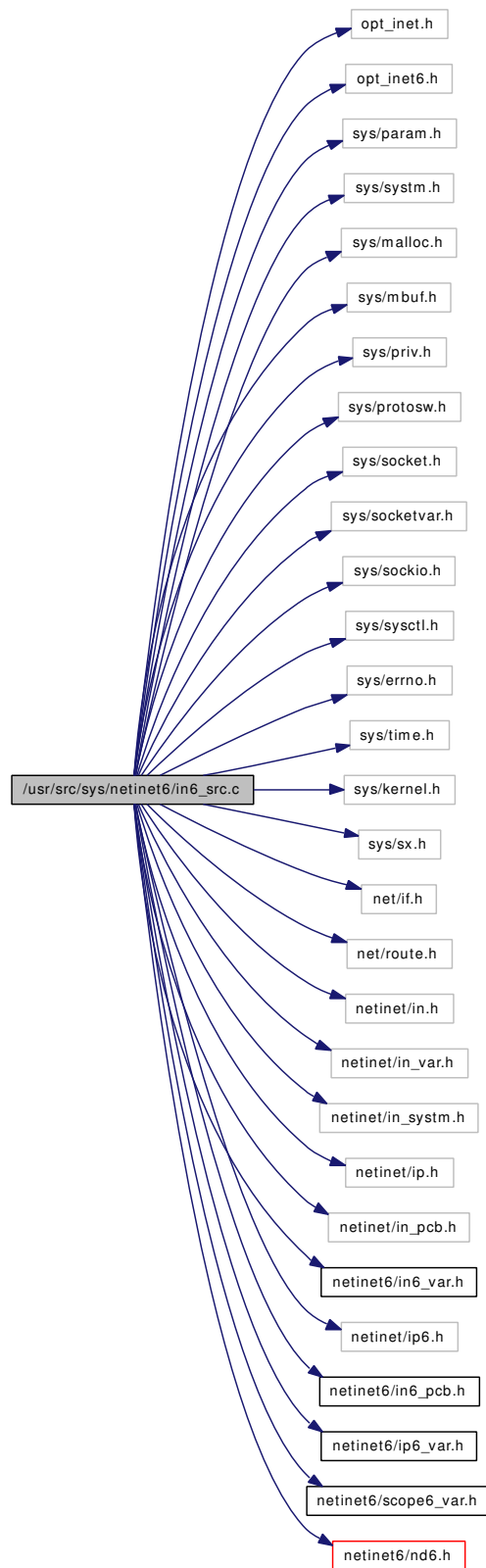
7.32.3.6 **int** `rtq_toomany` = 128 [static]

Definition at line 229 of file `in6_rmx.c`.

7.33 /usr/src/sys/netinet6/in6_src.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sockio.h>
#include <sys/sysctl.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/sx.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/in_pcb.h>
#include <netinet6/in6_var.h>
#include <netinet/ip6.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet6/nd6.h>
```

Include dependency graph for in6_src.c:



Data Structures

- struct [walkarg](#)
- struct [addrsel_policyent](#)

Defines

- #define [ADDRSEL_LOCK_INIT](#)() `mtx_init(&addrsel_lock, "addrsel_lock", NULL, MTX_DEF)`
- #define [ADDRSEL_LOCK](#)() `mtx_lock(&addrsel_lock)`
- #define [ADDRSEL_UNLOCK](#)() `mtx_unlock(&addrsel_lock)`
- #define [ADDRSEL_LOCK_ASSERT](#)() `mtx_assert(&addrsel_lock, MA_OWNED)`
- #define [ADDRSEL_SXLOCK_INIT](#)() `sx_init(&addrsel_sxlock, "addrsel_sxlock")`
- #define [ADDRSEL_SLOCK](#)() `sx_slock(&addrsel_sxlock)`
- #define [ADDRSEL_SUNLOCK](#)() `sx_sunlock(&addrsel_sxlock)`
- #define [ADDRSEL_XLOCK](#)() `sx_xlock(&addrsel_sxlock)`
- #define [ADDRSEL_XUNLOCK](#)() `sx_xunlock(&addrsel_sxlock)`
- #define [ADDR_LABEL_NOTAPP](#) (-1)
- #define [REPLACE](#)(r)
- #define [NEXT](#)(r)
- #define [BREAK](#)(r)

Functions

- static int [selectroute](#) `__P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct ifnet **, struct rentry **, int, int))`
- static int [in6_selectif](#) `__P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *ro, struct ifnet **))`
- static struct [in6_addrpolicy](#) *[lookup_addrsel_policy](#) `__P ((struct sockaddr_in6 *))`
- static void [init_policy_queue](#) `__P ((void))`
- static int [add_addrsel_policyent](#) `__P ((struct in6_addrpolicy *))`
- static int [walk_addrsel_policy](#) `__P ((int (*)(struct in6_addrpolicy *, void *), void *)`
- static int [dump_addrsel_policyent](#) `__P ((struct in6_addrpolicy *, void *))`
- [in6_addr](#) * [in6_selectsrc](#) (struct [sockaddr_in6](#) *dstsock, struct [ip6_pktopts](#) *opts, struct [ip6_moptions](#) *mopts, struct [route_in6](#) *ro, struct [in6_addr](#) *laddr, struct [ifnet](#) **ifpp, int *errorp)
- static int [selectroute](#) (struct [sockaddr_in6](#) *dstsock, struct [ip6_pktopts](#) *opts, struct [ip6_moptions](#) *mopts, struct [route_in6](#) *ro, struct [ifnet](#) **retifp, struct [rentry](#) **retrt, int clone, int norouteok)
- static int [in6_selectif](#) (struct [sockaddr_in6](#) *dstsock, struct [ip6_pktopts](#) *opts, struct [ip6_moptions](#) *mopts, struct [route_in6](#) *ro, struct [ifnet](#) **retifp)
- int [in6_selectroute](#) (struct [sockaddr_in6](#) *dstsock, struct [ip6_pktopts](#) *opts, struct [ip6_moptions](#) *mopts, struct [route_in6](#) *ro, struct [ifnet](#) **retifp, struct [rentry](#) **retrt, int clone)
- int [in6_selectthlim](#) (struct [in6pcb](#) *in6p, struct [ifnet](#) *ifp)
- int [in6_pcbsetport](#) (struct [in6_addr](#) *laddr, struct [inpcb](#) *inpcb, struct [ucred](#) *cred)
- void [addrsel_policy_init](#) ()
- static struct [in6_addrpolicy](#) * [lookup_addrsel_policy](#) (struct [sockaddr_in6](#) *key)
- static int [in6_src_sysctl](#) (SYSCTL_HANDLER_ARGS)
- [SYSCTL_DECL](#) (_net_inet6_ip6)
- [SYSCTL_NODE](#) (_net_inet6_ip6, IPV6CTL_ADDRCTLPOLICY, [addrctlpolicy](#), CTLFLAG_RD, [in6_src_sysctl](#), "")
- int [in6_src_ioctl](#) (u_long cmd, [caddr_t](#) data)
- [TAILQ_HEAD](#) ([addrsel_policyhead](#), [addrsel_policyent](#))

- static void [init_policy_queue](#) ()
- static int [add_addrsel_policyent](#) (struct [in6_addrpolicy](#) *newpolicy)
- static int [delete_addrsel_policyent](#) (struct [in6_addrpolicy](#) *key)
- static int [walk_addrsel_policy](#) (int *callback, void *w)
- static int [dump_addrsel_policyent](#) (struct [in6_addrpolicy](#) *pol, void *arg)
- static struct [in6_addrpolicy](#) * [match_addrsel_policy](#) (struct [sockaddr_in6](#) *key)

Variables

- static struct mtx [addrsel_lock](#)
- static struct sx [addrsel_sxlock](#)
- [in6_addrpolicy](#) [defaultaddrpolicy](#)
- int [ip6_prefer_tempaddr](#) = 0
- [addrsel_policyhead](#) [addrsel_policytab](#)

7.33.1 Define Documentation

7.33.1.1 #define ADDR_LABEL_NOTAPP (-1)

Definition at line 110 of file [in6_src.c](#).

Referenced by [addrsel_policy_init](#)(), and [in6_src_ioctl](#)().

7.33.1.2 #define ADDRSEL_LOCK() mtx_lock(&addrsel_lock)

Definition at line 99 of file [in6_src.c](#).

Referenced by [add_addrsel_policyent](#)(), [delete_addrsel_policyent](#)(), and [lookup_addrsel_policy](#)().

7.33.1.3 #define ADDRSEL_LOCK_ASSERT() mtx_assert(&addrsel_lock, MA_OWNED)

Definition at line 101 of file [in6_src.c](#).

7.33.1.4 #define ADDRSEL_LOCK_INIT() mtx_init(&addrsel_lock, "addrsel_lock", NULL, MTX_DEF)

Definition at line 98 of file [in6_src.c](#).

Referenced by [addrsel_policy_init](#)().

7.33.1.5 #define ADDRSEL_SLOCK() sx_slock(&addrsel_sxlock)

Definition at line 105 of file [in6_src.c](#).

Referenced by [walk_addrsel_policy](#)().

7.33.1.6 #define ADDRSEL_SUNLOCK() sx_sunlock(&addrsel_sxlock)

Definition at line 106 of file [in6_src.c](#).

Referenced by [walk_addrsel_policy](#)().

7.33.1.7 #define ADDRSEL_SXLOCK_INIT() sx_init(&addrsel_sxlock, "addrsel_sxlock")

Definition at line 104 of file in6_src.c.

Referenced by addrsel_policy_init().

7.33.1.8 #define ADDRSEL_UNLOCK() mtx_unlock(&addrsel_lock)

Definition at line 100 of file in6_src.c.

Referenced by add_addrsel_policyent(), delete_addrsel_policyent(), and lookup_addrsel_policy().

7.33.1.9 #define ADDRSEL_XLOCK() sx_xlock(&addrsel_sxlock)

Definition at line 107 of file in6_src.c.

Referenced by add_addrsel_policyent(), and delete_addrsel_policyent().

7.33.1.10 #define ADDRSEL_XUNLOCK() sx_xunlock(&addrsel_sxlock)

Definition at line 108 of file in6_src.c.

Referenced by add_addrsel_policyent(), and delete_addrsel_policyent().

7.33.1.11 #define BREAK(r)**Value:**

```
do { \
    if ((r) < sizeof(ip6stat.ip6s_sources_rule) / \
        sizeof(ip6stat.ip6s_sources_rule[0])) /* check for safety */ \
        ip6stat.ip6s_sources_rule[(r)]++; \
    goto out; /* XXX: we can't use 'break' here */ \
} while(0)
```

Definition at line 151 of file in6_src.c.

Referenced by in6_selectsrc().

7.33.1.12 #define NEXT(r)**Value:**

```
do { \
    if ((r) < sizeof(ip6stat.ip6s_sources_rule) / \
        sizeof(ip6stat.ip6s_sources_rule[0])) /* check for safety */ \
        ip6stat.ip6s_sources_rule[(r)]++; \
    /* printf("in6_selectsrc: keep %s against %s by %d\n", ia_best ? ip6_sprintf(&ia_best->ia_addr.sin
        goto next; /* XXX: we can't use 'continue' here */ \
} while(0)
```

Definition at line 144 of file in6_src.c.

7.33.1.13 #define REPLACE(r)

Value:

```
do { \
    if ((r) < sizeof(ip6stat.ip6s_sources_rule) / \
        sizeof(ip6stat.ip6s_sources_rule[0])) /* check for safety */ \
        ip6stat.ip6s_sources_rule[(r)]++; \
    /* printf("in6_selectsrc: replace %s with %s by %d\n", ia_best ? ip6_sprintf(&ia_best->ia_addr.sin
    goto replace; \
} while(0)
```

Definition at line 137 of file in6_src.c.

7.33.2 Function Documentation

7.33.2.1 static int dump_addrsel_policyent __P((struct in6_addrpolicy *, void *)) [static]

7.33.2.2 static int walk_addrsel_policy __P((int*)(struct in6_addrpolicy *, void *), void *) [static]

7.33.2.3 static int delete_addrsel_policyent __P((struct in6_addrpolicy *)) [static]

7.33.2.4 static void init_policy_queue __P((void)) [static]

7.33.2.5 static struct in6_addrpolicy *match_addrsel_policy __P((struct sockaddr_in6 *)) [static]

7.33.2.6 static int in6_selectif __P((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *ro, struct ifnet **)) [static]

7.33.2.7 static int selectroute __P((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct ifnet **, struct rtenry **, int, int)) [static]

7.33.2.8 static int add_addrsel_policyent (struct in6_addrpolicy *newpolicy) [static]

Definition at line 964 of file in6_src.c.

References ADDRSEL_LOCK, ADDRSEL_UNLOCK, ADDRSEL_XLOCK, ADDRSEL_XUNLOCK, and IN6_ARE_ADDR_EQUAL.

Referenced by in6_src_ioctl().

7.33.2.9 void addrsel_policy_init ()

Definition at line 851 of file in6_src.c.

References ADDR_LABEL_NOTAPP, ADDRSEL_LOCK_INIT, ADDRSEL_SXLOCK_INIT, default_addrpolicy, and init_policy_queue().

Here is the call graph for this function:



7.33.2.10 `static int delete_addrsel_policyent (struct in6_addrpolicy * key) [static]`

Definition at line 1000 of file `in6_src.c`.

References `ADDRSEL_LOCK`, `ADDRSEL_UNLOCK`, `ADDRSEL_XLOCK`, `ADDRSEL_XUNLOCK`, and `IN6_ARE_ADDR_EQUAL`.

Referenced by `in6_src_ioctl()`.

7.33.2.11 `static int dump_addrsel_policyent (struct in6_addrpolicy * pol, void * arg) [static]`

Definition at line 1050 of file `in6_src.c`.

References `walkarg::w_req`.

Referenced by `in6_src_sysctl()`.

7.33.2.12 `int in6_pcbsetport (struct in6_addr * laddr, struct inpcb * inp, struct ucred * cred)`

Definition at line 753 of file `in6_src.c`.

References `in6_pcblookup_local()`, and `in6addr_any`.

Referenced by `in6_pcbbind()`, and `udp6_output()`.

Here is the call graph for this function:



7.33.2.13 `int in6_selecthlim (struct in6pcb * in6p, struct ifnet * ifp)`

Definition at line 720 of file `in6_src.c`.

References `IN6_IS_ADDR_UNSPECIFIED`, `ip6_defhlim`, and `ND_IFINFO`.

Referenced by `rip6_output()`, and `udp6_output()`.

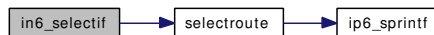
7.33.2.14 `static int in6_selectif (struct sockaddr_in6 * dstsock, struct ip6_pktopts * opts, struct ip6_moptions * mopts, struct route_in6 * ro, struct ifnet ** retifp) [static]`

Definition at line 635 of file `in6_src.c`.

References `selectroute()`.

Referenced by `in6_selectsrc()`.

Here is the call graph for this function:



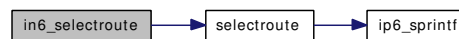
7.33.2.15 `int in6_selectroute (struct sockaddr_in6 * dstsock, struct ip6_pktopts * opts, struct ip6_moptions * mopts, struct route_in6 * ro, struct ifnet ** retifp, struct rentry ** retrt, int clone)`

Definition at line 699 of file in6_src.c.

References selectroute().

Referenced by ip6_output().

Here is the call graph for this function:



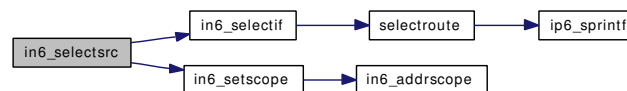
7.33.2.16 `struct in6_addr* in6_selectsrc (struct sockaddr_in6 * dstsock, struct ip6_pktopts * opts, struct ip6_moptions * mopts, struct route_in6 * ro, struct in6_addr * laddr, struct ifnet ** ifpp, int * errorp)`

Definition at line 159 of file in6_src.c.

References BREAK, in6_ifaddr::ia6_flags, in6_ifaddr::ia_addr, in6_ifaddr::ia_next, IFA6_IS_DEPRECATED, IN6_ARE_ADDR_EQUAL, IN6_IFF_ANYCAST, IN6_IFF_DETACHED, IN6_IFF_NOTREADY, IN6_IS_ADDR_UNSPECIFIED, in6_selectif(), in6_setscope(), ip6_use_deprecated, ip6_pktopts::ip6po_pktinfo, in6_pktinfo::ipi6_addr, and sockaddr_in6::sin6_addr.

Referenced by icmp6_reflect(), in6_pcblladdr(), nd6_na_output(), nd6_ns_output(), rip6_connect(), rip6_output(), and udp6_output().

Here is the call graph for this function:



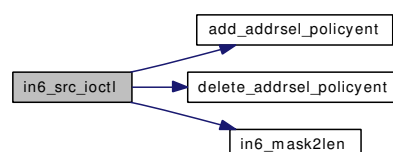
7.33.2.17 `int in6_src_ioctl (u_long cmd, caddr_t data)`

Definition at line 908 of file in6_src.c.

References add_addrsel_policyent(), in6_addrpolicy::addr, ADDR_LABEL_NOTAPP, in6_addrpolicy::addrmask, delete_addrsel_policyent(), in6_mask2len(), in6_addrpolicy::label, sockaddr_in6::sin6_addr, SIOCAADDRCTL_POLICY, SIOCADDRCTL_POLICY, and in6_addrpolicy::use.

Referenced by in6_control().

Here is the call graph for this function:

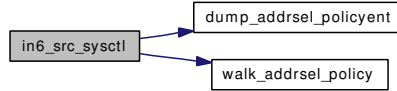


7.33.2.18 `static int in6_src_sysctl (SYSCALL_HANDLER_ARGS) [static]`

Definition at line 894 of file `in6_src.c`.

References `dump_addrsel_policyent()`, and `walk_addrsel_policy()`.

Here is the call graph for this function:



7.33.2.19 `static void init_policy_queue () [static]`

Definition at line 958 of file `in6_src.c`.

Referenced by `addrsel_policy_init()`.

7.33.2.20 `static struct in6_addrpolicy* lookup_addrsel_policy (struct sockaddr_in6 * key) [static]`

Definition at line 864 of file `in6_src.c`.

References `ADDRSEL_LOCK`, `ADDRSEL_UNLOCK`, `defaultaddrpolicy`, `match_addrsel_policy()`, and `in6_addrpolicy::use`.

Here is the call graph for this function:



7.33.2.21 `static struct in6_addrpolicy* match_addrsel_policy (struct sockaddr_in6 * key) [static]`

Definition at line 1063 of file `in6_src.c`.

Referenced by `lookup_addrsel_policy()`.

7.33.2.22 `static int selectroute (struct sockaddr_in6 * dstsock, struct ip6_pktopts * opts, struct ip6_moptions * mopts, struct route_in6 * ro, struct ifnet ** retifp, struct rentry ** retrt, int clone, int norouteok) [static]`

Definition at line 433 of file `in6_src.c`.

References `ip6_moptions::im6o_multicast_ifp`, `IN6_ARE_ADDR_EQUAL`, `IN6_IS_ADDR_LOOPBACK`, `IN6_IS_ADDR_MULTICAST`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `ip6_pktopts::ip6po_pktinfo`, `in6_pktinfo::ipi6_ifindex`, `satosin6`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, and `sockaddr_in6::sin6_scope_id`.

Referenced by `in6_selectif()`, and `in6_selectroute()`.

Here is the call graph for this function:



7.33.2.23 `SYSCTL_DECL (_net_inet6_ip6)`

7.33.2.24 `SYSCTL_NODE (_net_inet6_ip6, IPV6CTL_ADDRCTLPOLICY, addrcctlpolicy, CTLFLAG_RD, in6_src_sysctl, "")`

7.33.2.25 `TAILQ_HEAD (addrsel_policyhead, addrsel_policyent)`

7.33.2.26 `static int walk_addrsel_policy (int * callback, void * w)` `[static]`

Definition at line 1031 of file `in6_src.c`.

References `ADDRSEL_SLOCK`, and `ADDRSEL_SUNLOCK`.

Referenced by `in6_src_sysctl()`.

7.33.3 Variable Documentation

7.33.3.1 `struct mtx addrsel_lock` `[static]`

Definition at line 97 of file `in6_src.c`.

7.33.3.2 `struct addrsel_policyhead addrsel_policytab`

Definition at line 955 of file `in6_src.c`.

7.33.3.3 `struct sx addrsel_sxlock` `[static]`

Definition at line 103 of file `in6_src.c`.

7.33.3.4 `struct in6_addrpolicy defaultaddrpolicy`

Definition at line 111 of file `in6_src.c`.

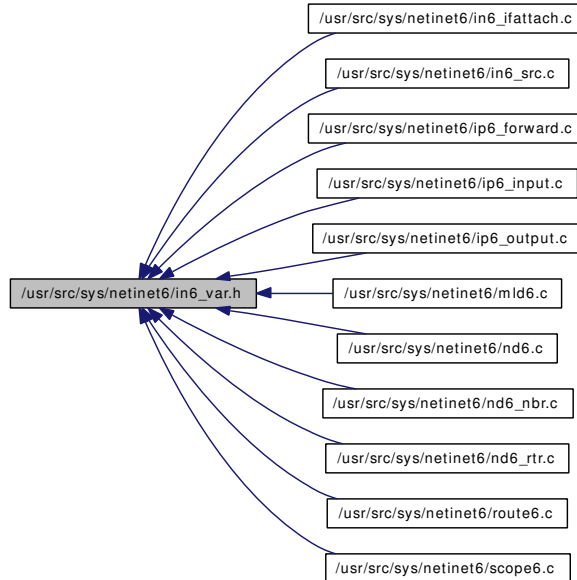
Referenced by `addrsel_policy_init()`, and `lookup_addrsel_policy()`.

7.33.3.5 `int ip6_prefer_tempaddr = 0`

Definition at line 113 of file `in6_src.c`.

7.34 /usr/src/sys/netinet6/in6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [in6_addrlifetime](#)
- struct [in6_ifextra](#)
- struct [in6_ifaddr](#)
- struct [in6_addrpolicy](#)
- struct [in6_ifstat](#)
- struct [icmp6_ifstat](#)
- struct [in6_ifreq](#)
- struct [in6_aliasreq](#)
- struct [in6_prflags](#)
- struct [in6_prflags::prf_ra](#)
- struct [in6_prflags::prf_rr](#)
- struct [in6_prefixreq](#)
- struct [in6_rrenumreq](#)
- struct [in6_rrenumreq::irr_raflagmask](#)
- struct [in6_multi_mship](#)
- struct [in6_multi](#)
- struct [in6_multistep](#)

Defines

- #define [ia_ifp](#) ia_ifa.ifa_ifp
- #define [ia_flags](#) ia_ifa.ifa_flags
- #define [IN6_PREFIX_ND](#) 1
- #define [IN6_PREFIX_RR](#) 2

- #define PR_ORIG_RA 0
- #define PR_ORIG_RR 1
- #define PR_ORIG_STATIC 2
- #define PR_ORIG_KERNEL 3
- #define ipr_raf_onlink ipr_flags.prf_ra.onlink
- #define ipr_raf_auto ipr_flags.prf_ra.autonomous
- #define ipr_statef_onlink ipr_flags.prf_state.onlink
- #define ipr_rrf_decrvalid ipr_flags.prf_rr.decrvalid
- #define ipr_rrf_decrprefd ipr_flags.prf_rr.decrprefd
- #define irr_raf_mask_onlink irr_raflagmask.onlink
- #define irr_raf_mask_auto irr_raflagmask.autonomous
- #define irr_raf_mask_reserved irr_raflagmask.reserved
- #define irr_raf_onlink irr_flags.prf_ra.onlink
- #define irr_raf_auto irr_flags.prf_ra.autonomous
- #define irr_statef_onlink irr_flags.prf_state.onlink
- #define irr_rrf irr_flags.prf_rr
- #define irr_rrf_decrvalid irr_flags.prf_rr.decrvalid
- #define irr_rrf_decrprefd irr_flags.prf_rr.decrprefd
- #define IA6_IN6(ia) (&((ia) → ia_addr.sin6_addr))
- #define IA6_DSTIN6(ia) (&((ia) → ia_dstaddr.sin6_addr))
- #define IA6_MASKIN6(ia) (&((ia) → ia_prefixmask.sin6_addr))
- #define IA6_SIN6(ia) (&((ia) → ia_addr))
- #define IA6_DSTSIN6(ia) (&((ia) → ia_dstaddr))
- #define IFA_IN6(x) (&((struct sockaddr_in6 *) (x) → ifa_addr) → sin6_addr)
- #define IFA_DSTIN6(x) (&((struct sockaddr_in6 *) (x) → ifa_dstaddr) → sin6_addr)
- #define IFPR_IN6(x) (&((struct sockaddr_in6 *) (x) → ifpr_prefix) → sin6_addr)
- #define IN6_ARE_MASKED_ADDR_EQUAL(d, a, m)
- #define SIOCSIFADDR_IN6_IOW('i', 12, struct in6_ifreq)
- #define SIOCGIFADDR_IN6_IOWR('i', 33, struct in6_ifreq)
- #define SIOCSIFDSTADDR_IN6_IOW('i', 14, struct in6_ifreq)
- #define SIOCSIFNETMASK_IN6_IOW('i', 22, struct in6_ifreq)
- #define SIOCGIFDSTADDR_IN6_IOWR('i', 34, struct in6_ifreq)
- #define SIOCGIFNETMASK_IN6_IOWR('i', 37, struct in6_ifreq)
- #define SIOCDFADDR_IN6_IOW('i', 25, struct in6_ifreq)
- #define SIOCAIFADDR_IN6_IOW('i', 26, struct in6_aliasreq)
- #define SIOCSIFPHYADDR_IN6_IOW('i', 70, struct in6_aliasreq)
- #define SIOCGIFSRCADDR_IN6_IOWR('i', 71, struct in6_ifreq)
- #define SIOCGIFPDSTADDR_IN6_IOWR('i', 72, struct in6_ifreq)
- #define SIOCGIFAFLAG_IN6_IOWR('i', 73, struct in6_ifreq)
- #define SIOCGDRLST_IN6_IOWR('i', 74, struct in6_drlst)
- #define SIOCGPRLST_IN6_IOWR('i', 75, struct in6_oprlist)
- #define OSIOCGIFINFO_IN6_IOWR('i', 76, struct in6_ondireq)
- #define SIOCGIFINFO_IN6_IOWR('i', 108, struct in6_ndireq)
- #define SIOCSIFINFO_IN6_IOWR('i', 109, struct in6_ndireq)
- #define SIOCSNDFLUSH_IN6_IOWR('i', 77, struct in6_ifreq)
- #define SIOCGNBRINFO_IN6_IOWR('i', 78, struct in6_nbrinfo)
- #define SIOCSPFXFLUSH_IN6_IOWR('i', 79, struct in6_ifreq)
- #define SIOCSRTRFLUSH_IN6_IOWR('i', 80, struct in6_ifreq)
- #define SIOCGIFALIFETIME_IN6_IOWR('i', 81, struct in6_ifreq)
- #define SIOCSIFALIFETIME_IN6_IOWR('i', 82, struct in6_ifreq)

- #define `SIOCGIFSTAT_IN6_IOWR('i', 83, struct in6_ifreq)`
- #define `SIOCGIFSTAT_ICMP6_IOWR('i', 84, struct in6_ifreq)`
- #define `SIOCSDEFIFACE_IN6_IOWR('i', 85, struct in6_ndifreq)`
- #define `SIOCGDEFIFACE_IN6_IOWR('i', 86, struct in6_ndifreq)`
- #define `SIOCSIFINFO_FLAGS_IOWR('i', 87, struct in6_ndireq)`
- #define `SIOCSCOPE6_IOW('i', 88, struct in6_ifreq)`
- #define `SIOCGSCOPE6_IOWR('i', 89, struct in6_ifreq)`
- #define `SIOCGSCOPE6DEF_IOWR('i', 90, struct in6_ifreq)`
- #define `SIOCSIFPREFIX_IN6_IOW('i', 100, struct in6_prefixreq)`
- #define `SIOCGIFPREFIX_IN6_IOWR('i', 101, struct in6_prefixreq)`
- #define `SIOCDFPREFIX_IN6_IOW('i', 102, struct in6_prefixreq)`
- #define `SIOCAIFPREFIX_IN6_IOW('i', 103, struct in6_rrenumreq)`
- #define `SIOCCIFPREFIX_IN6`
- #define `SIOCSGIFPREFIX_IN6`
- #define `SIOCGETSGCNT_IN6`
- #define `SIOCGETMIFCNT_IN6`
- #define `SIOCAADDRCTL_POLICY_IOW('u', 108, struct in6_addrpolicy)`
- #define `SIOCADDRCTL_POLICY_IOW('u', 109, struct in6_addrpolicy)`
- #define `IN6_IFF_ANYCAST 0x01`
- #define `IN6_IFF_TENTATIVE 0x02`
- #define `IN6_IFF_DUPLICATED 0x04`
- #define `IN6_IFF_DETACHED 0x08`
- #define `IN6_IFF_DEPRECATED 0x10`
- #define `IN6_IFF_NODAD 0x20`
- #define `IN6_IFF_AUTOCONF 0x40`
- #define `IN6_IFF_TEMPORARY 0x80`
- #define `IN6_IFF_NOPFX 0x8000`
- #define `IN6_IFF_NOTREADY (IN6_IFF_TENTATIVE|IN6_IFF_DUPLICATED)`
- #define `IN6_ARE_SCOPE_CMP(a, b) ((a)-(b))`
- #define `IN6_ARE_SCOPE_EQUAL(a, b) ((a)==(b))`
- #define `in6_ifstat_inc(ifp, tag)`
- #define `IFP_TO_IA6(ifp, ia)`
- #define `IN6M_TIMER_UNDEF -1`
- #define `IN6_IFAUPDATE_DADDELAY 0x1`
- #define `IN6_LOOKUP_MULTI(addr, ifp, in6m)`
- #define `IN6_NEXT_MULTI(step, in6m)`
- #define `IN6_FIRST_MULTI(step, in6m)`

Functions

- `LIST_HEAD (in6_multihead, in6_multi) in6_multihead`
- `in6_multi *in6_addmulti __P ((struct in6_addr *, struct ifnet *, int *, int))`
- `void in6_delmulti __P ((struct in6_multi *))`
- `in6_multi_mship * in6_joiningroup (struct ifnet *, struct in6_addr *, int *, int)`
- `int in6_leavegroup (struct in6_multi_mship *)`
- `int in6_mask2len __P ((struct in6_addr *, u_char *))`
- `int in6_control __P ((struct socket *, u_long, caddr_t, struct ifnet *, struct thread *))`
- `int in6_update_ifa __P ((struct ifnet *, struct in6_aliasreq *, struct in6_ifaddr *, int))`
- `void in6_purgeaddr __P ((struct ifaddr *))`
- `int in6if_do_dad __P ((struct ifnet *))`

- void in6_savemkludge __P ((struct in6_ifaddr *))
- void in6_domifdetach __P ((struct ifnet *, void *))
- void in6_setmaxmtu __P ((void))
- void in6_restoremkludge __P ((struct in6_ifaddr *, struct ifnet *))
- in6_ifaddr *in6ifa_ifpforlinklocal __P ((struct ifnet *, int))
- in6_ifaddr *in6ifa_ifpwithaddr __P ((struct ifnet *, struct in6_addr *))
- char *ip6_sprintf __P ((char *, const struct in6_addr *))
- int in6_addr2zoneid __P ((struct ifnet *, struct in6_addr *, u_int32_t *))
- int in6_matchlen __P ((struct in6_addr *, struct in6_addr *))
- int in6_are_prefix_equal __P ((struct in6_addr *, struct in6_addr *, int))
- void in6_prefixlen2mask __P ((struct in6_addr *, int))
- int in6_prefix_ioctl __P ((struct socket *, u_long, caddr_t, struct ifnet *))
- int in6_prefix_add_ifid __P ((int, struct in6_ifaddr *))
- void in6_ifremloop (struct ifaddr *)
- void in6_ifaddloop (struct ifaddr *)
- int in6_is_addr_deprecated __P ((struct sockaddr_in6 *))
- int in6_src_ioctl __P ((u_long, caddr_t))

Variables

- in6_ifaddr * in6_ifaddr
- icmp6stat icmp6stat
- in6_addr zeroin6_addr
- u_char inet6ctlerrmap []
- unsigned long in6_maxmtu

7.34.1 Define Documentation

7.34.1.1 #define IA6_DSTIN6(ia) (&((ia) → ia_dstaddr.sin6_addr))

Definition at line 360 of file in6_var.h.

7.34.1.2 #define IA6_DSTSIN6(ia) (&((ia) → ia_dstaddr))

Definition at line 363 of file in6_var.h.

7.34.1.3 #define IA6_IN6(ia) (&((ia) → ia_addr.sin6_addr))

Definition at line 359 of file in6_var.h.

7.34.1.4 #define IA6_MASKIN6(ia) (&((ia) → ia_prefixmask.sin6_addr))

Definition at line 361 of file in6_var.h.

7.34.1.5 #define IA6_SIN6(ia) (&((ia) → ia_addr))

Definition at line 362 of file in6_var.h.

7.34.1.6 #define ia_flags ia_ifa.ifa_flags

Definition at line 101 of file in6_var.h.

7.34.1.7 #define ia_ifp ia_ifa.ifa_ifp

Definition at line 100 of file in6_var.h.

7.34.1.8 #define IFA_DSTIN6(x) (&((struct sockaddr_in6 *) (x) → ifa_dstaddr) → sin6_addr)

Definition at line 365 of file in6_var.h.

7.34.1.9 #define IFA_IN6(x) (&((struct sockaddr_in6 *) (x) → ifa_addr) → sin6_addr)

Definition at line 364 of file in6_var.h.

Referenced by in6_ifawithifp(), in6_ifremloop(), in6_lifaddr_ioctl(), in6ifa_ifpforlinklocal(), in6ifa_ifpwithaddr(), and nd6_ns_input().

7.34.1.10 #define IFP_TO_IA6(ifp, ia)**Value:**

```

/* struct ifnet *ifp; */
/* struct in6_ifaddr *ia; */
do {
    struct ifaddr *ifa;
    TAILQ_FOREACH(ifa, &(ifp)→if_addrlist, ifa_list) {
        if (ifa→ifa_addr→sa_family == AF_INET6)
            break;
    }
    (ia) = (struct in6_ifaddr *)ifa;
} while (/*CONSTCOND*/ 0)

```

Definition at line 491 of file in6_var.h.

Referenced by mld6_input().

7.34.1.11 #define IFPR_IN6(x) (&((struct sockaddr_in6 *) (x) → ifpr_prefix) → sin6_addr)

Definition at line 367 of file in6_var.h.

7.34.1.12 #define IN6_ARE_MASKED_ADDR_EQUAL(d, a, m)**Value:**

```

(
    (((d)→s6_addr32[0] ^ (a)→s6_addr32[0]) & (m)→s6_addr32[0]) == 0 && \
    (((d)→s6_addr32[1] ^ (a)→s6_addr32[1]) & (m)→s6_addr32[1]) == 0 && \
    (((d)→s6_addr32[2] ^ (a)→s6_addr32[2]) & (m)→s6_addr32[2]) == 0 && \
    (((d)→s6_addr32[3] ^ (a)→s6_addr32[3]) & (m)→s6_addr32[3]) == 0 )

```

Definition at line 370 of file in6_var.h.

Referenced by in6_localaddr(), and nd6_is_new_addr_neighbor().

7.34.1.13 #define IN6_ARE_SCOPE_CMP(a, b) ((a)-(b))

Definition at line 465 of file in6_var.h.

7.34.1.14 #define IN6_ARE_SCOPE_EQUAL(a, b) ((a)==(b))

Definition at line 466 of file in6_var.h.

7.34.1.15 #define IN6_FIRST_MULTI(step, in6m)**Value:**

```
/* struct in6_multistep step; */           \
/* struct in6_multi *in6m */             \
do { \
    (step).i_in6m = LIST_FIRST(&in6_multihead); \
    IN6_NEXT_MULTI((step), (in6m)); \
} while(0)
```

Definition at line 581 of file in6_var.h.

7.34.1.16 #define IN6_IFAUPDATE_DADDELAY 0x1

Definition at line 530 of file in6_var.h.

Referenced by in6_ifadd(), in6_ifattach_linklocal(), and in6_update_ifa().

7.34.1.17 #define IN6_IFF_ANYCAST 0x01

Definition at line 447 of file in6_var.h.

Referenced by icmp6_redirect_output(), icmp6_reflect(), in6_ifawithifp(), in6_pcbbind(), in6_selectsrc(), mld6_sendpkt(), nd6_dad_start(), nd6_ns_input(), nd6_prefix_onlink(), ni6_addrs(), ni6_store_addrs(), prelist_update(), and rip6_bind().

7.34.1.18 #define IN6_IFF_AUTOCONF 0x40

Definition at line 455 of file in6_var.h.

Referenced by in6_control(), in6_unlink_ifa(), nd6_ioctl(), pfxlist_onlink_check(), prelist_update(), and regen_tmpaddr().

7.34.1.19 #define IN6_IFF_DEPRECATED 0x10

Definition at line 451 of file in6_var.h.

Referenced by in6_ifawithifp(), in6_is_addr_deprecated(), in6_update_ifa(), nd6_timer(), ni6_store_addrs(), and rip6_bind().

7.34.1.20 #define IN6_IFF_DETACHED 0x08

Definition at line 450 of file in6_var.h.

Referenced by in6_ifawithifp(), in6_pcbbind(), in6_selectsrc(), and rip6_bind().

7.34.1.21 #define IN6_IFF_DUPLICATED 0x04

Definition at line 449 of file in6_var.h.

Referenced by in6_update_ifa(), nd6_dad_duplicated(), nd6_dad_timer(), and nd6_ns_input().

7.34.1.22 #define IN6_IFF_NODAD 0x20

Definition at line 452 of file in6_var.h.

Referenced by in6_ifattach_loopback(), and in6_update_ifa().

7.34.1.23 #define IN6_IFF_NOPFX 0x8000

Definition at line 457 of file in6_var.h.

Referenced by in6_ifattach_loopback().

7.34.1.24 #define IN6_IFF_NOTREADY (IN6_IFF_TENTATIVE|IN6_IFF_DUPLICATED)

Definition at line 462 of file in6_var.h.

Referenced by icmp6_redirect_output(), icmp6_reflect(), in6_ifawithifp(), in6_pcbbind(), in6_selectsrc(), ip6_input(), mld6_sendpkt(), nd6_ns_input(), nd6_prefix_onlink(), and rip6_bind().

7.34.1.25 #define IN6_IFF_TEMPORARY 0x80

Definition at line 456 of file in6_var.h.

Referenced by nd6_timer(), ni6_addrs(), ni6_input(), ni6_store_addrs(), prelist_update(), and regen_tmpaddr().

7.34.1.26 #define IN6_IFF_TENTATIVE 0x02

Definition at line 448 of file in6_var.h.

Referenced by in6_if_up(), in6_update_ifa(), nd6_dad_duplicated(), nd6_dad_start(), nd6_dad_timer(), nd6_na_input(), and nd6_ns_input().

7.34.1.27 #define in6_ifstat_inc(ifp, tag)

Value:

```
do {
    if (ifp)
        ((struct in6_ifextra *) ((ifp)->if_afdata[AF_INET6]))->in6_ifstat->tag++; \
} while (/*CONSTCOND*/ 0)
```

Definition at line 473 of file in6_var.h.

Referenced by frag6_input(), ip6_forward(), ip6_input(), and ip6_output().

7.34.1.28 #define IN6_LOOKUP_MULTI(addr, ifp, in6m)

Value:

```

/* struct in6_addr addr; */           \
/* struct ifnet *ifp; */             \
/* struct in6_multi *in6m; */       \
do { \
    struct ifmultiaddr *ifma; \
    IF_ADDR_LOCK(ifp); \
    TAILQ_FOREACH(ifma, &(ifp->if_multiaddrs, ifma_link) { \
        if (ifma->ifma_addr->sa_family == AF_INET6 \
            && IN6_ARE_ADDR_EQUAL(&((struct sockaddr_in6 *)ifma->ifma_addr)->sin6_addr, \
                                   &(addr))) \
            break; \
        } \
        (in6m) = (struct in6_multi *) (ifma ? ifma->ifma_protospec : 0); \
        IF_ADDR_UNLOCK(ifp); \
    } while(0)

```

Definition at line 549 of file in6_var.h.

Referenced by in6_purgeaddr(), ip6_input(), ip6_output(), mld6_input(), and phyint_send().

7.34.1.29 #define IN6_NEXT_MULTI(step, in6m)

Value:

```

/* struct in6_multistep step; */     \
/* struct in6_multi *in6m; */       \
do { \
    if (((in6m) = (step).i_in6m) != NULL) \
        (step).i_in6m = LIST_NEXT((step).i_in6m, in6m_entry); \
    } while(0)

```

Definition at line 573 of file in6_var.h.

7.34.1.30 #define IN6_PREFIX_ND 1

Definition at line 274 of file in6_var.h.

7.34.1.31 #define IN6_PREFIX_RR 2

Definition at line 275 of file in6_var.h.

7.34.1.32 #define IN6M_TIMER_UNDEF -1

Definition at line 526 of file in6_var.h.

Referenced by mld6_input(), mld_stoptimer(), and mld_timeo().

7.34.1.33 #define ipr_raf_auto ipr_flags.prf_ra.autonomous

Definition at line 315 of file in6_var.h.

7.34.1.34 #define ipr_raf_onlink ipr_flags.prf_ra.onlink

Definition at line 314 of file in6_var.h.

7.34.1.35 #define ipr_rrf_decrprefd ipr_flags.prf_rr.decrprefd

Definition at line 320 of file in6_var.h.

7.34.1.36 #define ipr_rrf_decrvalid ipr_flags.prf_rr.decrvalid

Definition at line 319 of file in6_var.h.

7.34.1.37 #define ipr_statef_onlink ipr_flags.prf_state.onlink

Definition at line 317 of file in6_var.h.

7.34.1.38 #define irr_raf_auto irr_flags.prf_ra.autonomous

Definition at line 347 of file in6_var.h.

7.34.1.39 #define irr_raf_mask_auto irr_raflagmask.autonomous

Definition at line 343 of file in6_var.h.

7.34.1.40 #define irr_raf_mask_onlink irr_raflagmask.onlink

Definition at line 342 of file in6_var.h.

7.34.1.41 #define irr_raf_mask_reserved irr_raflagmask.reserved

Definition at line 344 of file in6_var.h.

7.34.1.42 #define irr_raf_onlink irr_flags.prf_ra.onlink

Definition at line 346 of file in6_var.h.

7.34.1.43 #define irr_rrf irr_flags.prf_rr

Definition at line 351 of file in6_var.h.

7.34.1.44 #define irr_rrf_decrprefd irr_flags.prf_rr.decrprefd

Definition at line 353 of file in6_var.h.

7.34.1.45 #define irr_rrf_decrvalid irr_flags.prf_rr.decrvalid

Definition at line 352 of file in6_var.h.

7.34.1.46 #define irr_statef_onlink irr_flags.prf_state.onlink

Definition at line 349 of file in6_var.h.

7.34.1.47 #define OSIOCGIFINFO_IN6_IOWR('i', 76, struct in6_ondireq)

Definition at line 407 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.48 #define PR_ORIG_KERNEL 3

Definition at line 312 of file in6_var.h.

7.34.1.49 #define PR_ORIG_RA 0

Definition at line 309 of file in6_var.h.

Referenced by nd6_ioctl(), and nd6_sysctl_prlist().

7.34.1.50 #define PR_ORIG_RR 1

Definition at line 310 of file in6_var.h.

7.34.1.51 #define PR_ORIG_STATIC 2

Definition at line 311 of file in6_var.h.

7.34.1.52 #define SIOCAADDRCTL_POLICY_IOW('u', 108, struct in6_addrpolicy)

Definition at line 444 of file in6_var.h.

Referenced by in6_control(), and in6_src_ioctl().

7.34.1.53 #define SIOCAIFADDR_IN6_IOW('i', 26, struct in6_aliasreq)

Definition at line 393 of file in6_var.h.

Referenced by in6_control(), and in6_lifaddr_ioctl().

7.34.1.54 #define SIOCAIFPREFIX_IN6_IOW('i', 103, struct in6_rrenumreq)

Definition at line 433 of file in6_var.h.

Referenced by in6_control().

7.34.1.55 #define SIOCCIFPREFIX_IN6**Value:**

```
_IOW('i', 104, \
                                struct in6_rrenumreq)
```

Definition at line 434 of file in6_var.h.

Referenced by in6_control().

7.34.1.56 #define SIOCADDRCTL_POLICY_IOW('u', 109, struct in6_addrpolicy)

Definition at line 445 of file in6_var.h.

Referenced by in6_control(), and in6_src_ioctl().

7.34.1.57 #define SIOCIFADDR_IN6_IOW('i', 25, struct in6_ifreq)

Definition at line 392 of file in6_var.h.

Referenced by in6_control(), and in6_lifaddr_ioctl().

7.34.1.58 #define SIOCIFPREFIX_IN6_IOW('i', 102, struct in6_prefixreq)

Definition at line 432 of file in6_var.h.

Referenced by in6_control().

7.34.1.59 #define SIOCGDEFIFACE_IN6_IOWR('i', 86, struct in6_ndifreq)

Definition at line 422 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.60 #define SIOCGDRLST_IN6_IOWR('i', 74, struct in6_drlst)

Definition at line 401 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.61 #define SIOCGETMIFCNT_IN6**Value:**

```
_IOWR('u', 107, \
                                struct sioc_mif_req6)
```

Definition at line 441 of file in6_var.h.

Referenced by in6_control(), and mrt6_ioctl().

7.34.1.62 #define SIOCGETSGCNT_IN6

Value:

```
_IOWR('u', 106, \
                                struct sioc_sg_req6)
```

Definition at line 439 of file in6_var.h.

Referenced by in6_control(), and mrt6_ioctl().

7.34.1.63 #define SIOCGIFADDR_IN6 _IOWR('i', 33, struct in6_ifreq)

Definition at line 378 of file in6_var.h.

Referenced by in6_control().

7.34.1.64 #define SIOCGIFAFLAG_IN6 _IOWR('i', 73, struct in6_ifreq)

Definition at line 399 of file in6_var.h.

Referenced by in6_control().

7.34.1.65 #define SIOCGIFALIFETIME_IN6 _IOWR('i', 81, struct in6_ifreq)

Definition at line 416 of file in6_var.h.

Referenced by in6_control().

7.34.1.66 #define SIOCGIFDSTADDR_IN6 _IOWR('i', 34, struct in6_ifreq)

Definition at line 389 of file in6_var.h.

Referenced by in6_control().

7.34.1.67 #define SIOCGIFINFO_IN6 _IOWR('i', 108, struct in6_ndireq)

Definition at line 409 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.68 #define SIOCGIFNETMASK_IN6 _IOWR('i', 37, struct in6_ifreq)

Definition at line 390 of file in6_var.h.

Referenced by in6_control().

7.34.1.69 #define SIOCGIFPDSTADDR_IN6_IOWR('i', 72, struct in6_ifreq)

Definition at line 397 of file in6_var.h.

Referenced by in6_control().

7.34.1.70 #define SIOCGIFPREFIX_IN6_IOWR('i', 101, struct in6_prefixreq)

Definition at line 431 of file in6_var.h.

Referenced by in6_control().

7.34.1.71 #define SIOCGIFPSRCADDR_IN6_IOWR('i', 71, struct in6_ifreq)

Definition at line 396 of file in6_var.h.

Referenced by in6_control().

7.34.1.72 #define SIOCGIFSTAT_ICMP6_IOWR('i', 84, struct in6_ifreq)

Definition at line 419 of file in6_var.h.

Referenced by in6_control().

7.34.1.73 #define SIOCGIFSTAT_IN6_IOWR('i', 83, struct in6_ifreq)

Definition at line 418 of file in6_var.h.

Referenced by in6_control().

7.34.1.74 #define SIOCGNBRINFO_IN6_IOWR('i', 78, struct in6_nbrinfo)

Definition at line 412 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.75 #define SIOCGPRLST_IN6_IOWR('i', 75, struct in6_oprlist)

Definition at line 404 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.76 #define SIOCGSCOPE6_IOWR('i', 89, struct in6_ifreq)

Definition at line 427 of file in6_var.h.

Referenced by in6_control().

7.34.1.77 #define SIOCGSCOPE6DEF_IOWR('i', 90, struct in6_ifreq)

Definition at line 428 of file in6_var.h.

Referenced by in6_control().

7.34.1.78 #define SIOCSDEFIFACE_IN6_IOWR('i', 85, struct in6_ndifreq)

Definition at line 421 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.79 #define SIOCSGIFPREFIX_IN6**Value:**

```
_IOW('i', 105, \
                                struct in6_rrenumreq)
```

Definition at line 436 of file in6_var.h.

Referenced by in6_control().

7.34.1.80 #define SIOCSIFADDR_IN6_IOW('i', 12, struct in6_ifreq)

Definition at line 377 of file in6_var.h.

Referenced by in6_control().

7.34.1.81 #define SIOCSIFALIFETIME_IN6_IOWR('i', 82, struct in6_ifreq)

Definition at line 417 of file in6_var.h.

Referenced by in6_control().

7.34.1.82 #define SIOCSIFDSTADDR_IN6_IOW('i', 14, struct in6_ifreq)

Definition at line 385 of file in6_var.h.

Referenced by in6_control().

7.34.1.83 #define SIOCSIFINFO_FLAGS_IOWR('i', 87, struct in6_ndireq)

Definition at line 424 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.84 #define SIOCSIFINFO_IN6_IOWR('i', 109, struct in6_ndireq)

Definition at line 410 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.85 #define SIOCSIFNETMASK_IN6_IOW('i', 22, struct in6_ifreq)

Definition at line 386 of file in6_var.h.

Referenced by in6_control().

7.34.1.86 #define SIOCSIFPHYADDR_IN6_IOW('i', 70, struct in6_aliasreq)

Definition at line 395 of file in6_var.h.

Referenced by in6_control().

7.34.1.87 #define SIOCSIFPREFIX_IN6_IOW('i', 100, struct in6_prefixreq)

Definition at line 430 of file in6_var.h.

Referenced by in6_control().

7.34.1.88 #define SIOCSNDFLUSH_IN6_IOWR('i', 77, struct in6_ifreq)

Definition at line 411 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.89 #define SIOCSPFXFLUSH_IN6_IOWR('i', 79, struct in6_ifreq)

Definition at line 413 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.90 #define SIOCSRTRFLUSH_IN6_IOWR('i', 80, struct in6_ifreq)

Definition at line 414 of file in6_var.h.

Referenced by in6_control(), and nd6_ioctl().

7.34.1.91 #define SIOCSSCOPE6_IOW('i', 88, struct in6_ifreq)

Definition at line 426 of file in6_var.h.

Referenced by in6_control().

7.34.2 Function Documentation

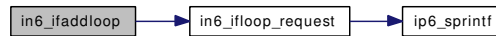
- 7.34.2.1 `int in6_src_ioctl __P ((u_long, caddr_t))`
- 7.34.2.2 `int in6_is_addr_deprecated __P ((struct sockaddr_in6 *))`
- 7.34.2.3 `void in6_prefix_remove_ifid __P ((int, struct in6_ifaddr *))`
- 7.34.2.4 `int in6_prefix_ioctl __P ((struct socket *, u_long, caddr_t, struct ifnet *))`
- 7.34.2.5 `void in6_prefixlen2mask __P ((struct in6_addr *, int))`
- 7.34.2.6 `int in6_are_prefix_equal __P ((struct in6_addr *, struct in6_addr *, int))`
- 7.34.2.7 `int in6_matchlen __P ((struct in6_addr *, struct in6_addr *))`
- 7.34.2.8 `int in6_addr2zoneid __P ((struct ifnet *, struct in6_addr *, u_int32_t *))`
- 7.34.2.9 `char* ip6_sprintf __P ((char *, const struct in6_addr *))`
- 7.34.2.10 `struct in6_ifaddr* in6ifa_ifpwithaddr __P ((struct ifnet *, struct in6_addr *))`
- 7.34.2.11 `struct in6_ifaddr* in6ifa_ifpforlinklocal __P ((struct ifnet *, int))`
- 7.34.2.12 `void in6_restoremkludge __P ((struct in6_ifaddr *, struct ifnet *))`
- 7.34.2.13 `void in6_setmaxmtu __P ((void))`
- 7.34.2.14 `void in6_domifdetach __P ((struct ifnet *, void *))`
- 7.34.2.15 `void in6_savemkludge __P ((struct in6_ifaddr *))`
- 7.34.2.16 `int in6if_do_dad __P ((struct ifnet *))`
- 7.34.2.17 `static void nd6_dad_na_input __P ((struct ifaddr *))`
- 7.34.2.18 `int in6_update_ifa __P ((struct ifnet *, struct in6_aliasreq *, struct in6_ifaddr *, int))`
- 7.34.2.19 `int in6_control __P ((struct socket *, u_long, caddr_t, struct ifnet *, struct thread *))`
- 7.34.2.20 `int in6_mask2len __P ((struct in6_addr *, u_char *))`
- 7.34.2.21 `void in6_delmulti __P ((struct in6_multi *))`
- 7.34.2.22 `struct in6_multi* in6_addmulti __P ((struct in6_addr *, struct ifnet *, int *, int))`
- 7.34.2.23 `void in6_ifaddloop (struct ifaddr *)`

Definition at line 209 of file in6.c.

References `in6_ifloop_request()`.

Referenced by `in6_ifinit()`.

Here is the call graph for this function:



7.34.2.24 void in6_ifremloop (struct ifaddr *)

Definition at line 229 of file in6.c.

References in6_ifaddr::ia_addr, in6_ifaddr::ia_next, IFA_IN6, IN6_ARE_ADDR_EQUAL, and sockaddr_in6::sin6_addr.

Referenced by in6_purgeaddr().

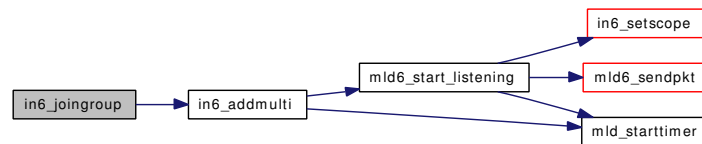
7.34.2.25 struct in6_multi_mship* in6_joingroup (struct ifnet *, struct in6_addr *, int *, int)

Definition at line 1817 of file in6.c.

References in6_addmulti().

Referenced by in6_update_ifa(), and ip6_setmoptions().

Here is the call graph for this function:

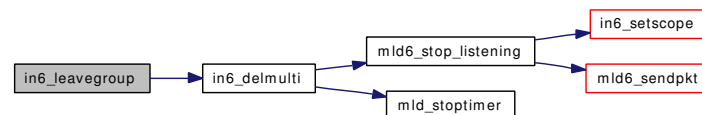


7.34.2.26 int in6_leavegroup (struct in6_multi_mship *)

Definition at line 1840 of file in6.c.

References in6_delmulti().

Here is the call graph for this function:



7.34.2.27 LIST_HEAD (in6_multihead, in6_multi)

7.34.3 Variable Documentation

7.34.3.1 struct icmp6stat icmp6stat

Definition at line 114 of file icmp6.c.

Referenced by `icmp6_error()`, `icmp6_input()`, `icmp6_mtudisc_update()`, `icmp6_notify_error()`, `icmp6_redirect_input()`, `icmp6_redirect_output()`, `mld6_input()`, `mld6_sendpkt()`, `nd6_na_input()`, `nd6_na_output()`, `nd6_ns_input()`, `nd6_ns_output()`, `nd6_options()`, `nd6_ra_input()`, `nd6_rs_input()`, and `rip6_output()`.

7.34.3.2 struct `in6_ifaddr*` `in6_ifaddr`

Definition at line 124 of file `ip6_input.c`.

Referenced by `ip6_input()`.

7.34.3.3 unsigned long `in6_maxmtu`

Definition at line 61 of file `in6_ifattach.c`.

Referenced by `in6_ifattach()`, `in6_setmaxmtu()`, and `nd6_setmtu0()`.

7.34.3.4 u_char `inet6ctlerrmap[]`

Definition at line 1580 of file `ip6_input.c`.

Referenced by `in6_pcbnotify()`, `rip6_ctlinput()`, `sctp6_ctlinput()`, and `udp6_ctlinput()`.

7.34.3.5 struct `in6_addr` `zeroin6_addr`

Definition at line 120 of file `in6_pcb.c`.

7.35 /usr/src/sys/netinet6/ip6.h File Reference

7.36 /usr/src/sys/netinet6/ip6_ecn.h File Reference

Functions

- void [ip6_ecn_ingress](#) (int, u_int32_t *, const u_int32_t *)
- int [ip6_ecn_egress](#) (int, const u_int32_t *, u_int32_t *)

7.36.1 Function Documentation

7.36.1.1 int [ip6_ecn_egress](#) (int, const u_int32_t *, u_int32_t *)

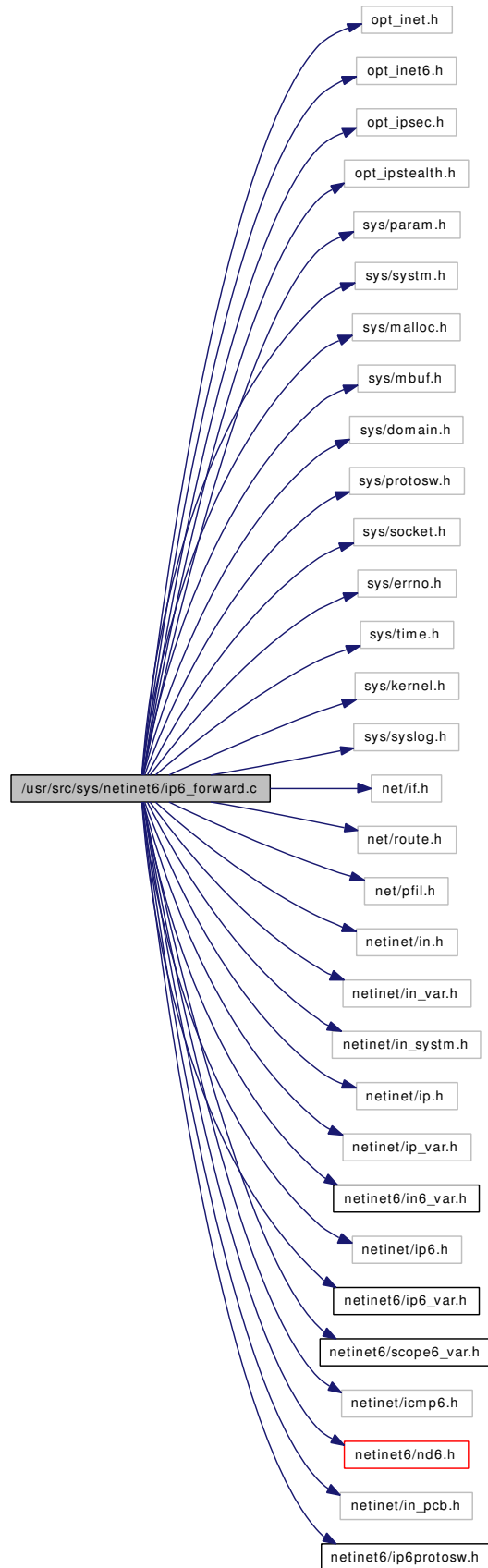
Referenced by [in6_gif_input\(\)](#).

7.36.1.2 void [ip6_een_ingress](#) (int, u_int32_t *, const u_int32_t *)

7.37 /usr/src/sys/netinet6/ip6_forward.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include "opt_ipstealth.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <net/pfil.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet6/in6_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet/icmp6.h>
#include <netinet6/nd6.h>
#include <netinet/in_pcb.h>
#include <netinet6/ip6protosw.h>
```

Include dependency graph for ip6_forward.c:



Functions

- void [ip6_forward](#) (struct mbuf *m, int srcrt)

Variables

- route_in6 [ip6_forward_rt](#)

7.37.1 Function Documentation

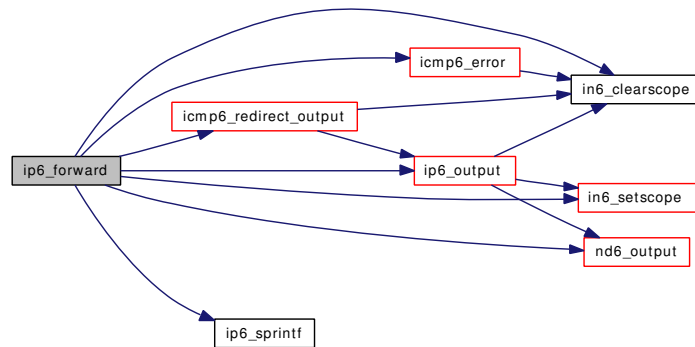
7.37.1.1 void ip6_forward (struct mbuf * m, int srcrt)

Definition at line 101 of file ip6_forward.c.

References ipsec_output_state::dst, icmp6_error(), icmp6_redirect_output(), IN6_ARE_ADDR_EQUAL, in6_clearscope(), in6_ifstat_inc, IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, IN6_LINKMTU, in6_setscope(), ipsecstat::in_polvio, INET6_ADDRSTRLEN, inet6_pfil_hook, ip6_forward_rt, ip6_log_interval, ip6_log_time, ip6_output(), ip6_sendredirects, ip6_sprintf(), ipsec6stat, IPSEC_DIR_OUTBOUND, IPSEC_MODE_ANY, IPSEC_MODE_TUNNEL, IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, IPV6_FORWARDING, nd6_output(), ipsecrequest::next, ipsecstat::out_inval, ipsecstat::out_polvio, secpolicy::policy, secpolicy::req, ipsecrequest::saidx, senderr, and sockaddr_in6::sin6_addr.

Referenced by ip6_input(), and ip6_rthdr0().

Here is the call graph for this function:



7.37.2 Variable Documentation

7.37.2.1 struct route_in6 ip6_forward_rt

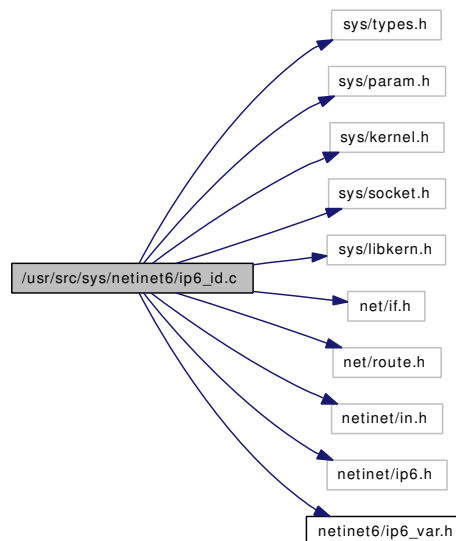
Definition at line 85 of file ip6_forward.c.

Referenced by frag6_slowtimo(), and ip6_forward().

7.38 /usr/src/sys/netinet6/ip6_id.c File Reference

```
#include <sys/types.h>
#include <sys/param.h>
#include <sys/kernel.h>
#include <sys/socket.h>
#include <sys/libkern.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
```

Include dependency graph for ip6_id.c:



Data Structures

- struct [randomtab](#)

Defines

- #define [INT32_MAX](#) 0x7fffffffU

Functions

- static u_int32_t [pmod](#) (u_int32_t, u_int32_t, u_int32_t)
- static void [initid](#) (struct [randomtab](#) *)
- static u_int32_t [randomid](#) (struct [randomtab](#) *)

- `u_int32_t ip6_randomid` (void)
- `u_int32_t ip6_randomflowlabel` (void)

Variables

- static struct `randomtab randomtab_32`
- static struct `randomtab randomtab_20`

7.38.1 Define Documentation

7.38.1.1 #define INT32_MAX 0x7fffffffU

Definition at line 102 of file `ip6_id.c`.

7.38.2 Function Documentation

7.38.2.1 static void initid (struct `randomtab *`) [static]

Definition at line 183 of file `ip6_id.c`.

References `randomtab::pfacts`, `pmod()`, `randomtab::ru_a`, `randomtab::ru_agen`, `randomtab::ru_b`, `randomtab::ru_bits`, `randomtab::ru_counter`, `randomtab::ru_g`, `randomtab::ru_gen`, `randomtab::ru_m`, `randomtab::ru_msb`, `randomtab::ru_n`, `randomtab::ru_out`, `randomtab::ru_reseed`, `randomtab::ru_seed`, `randomtab::ru_seed2`, and `randomtab::ru_x`.

Referenced by `randomid()`.

Here is the call graph for this function:



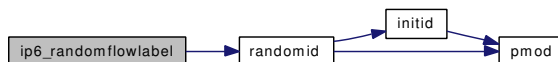
7.38.2.2 u_int32_t ip6_randomflowlabel (void)

Definition at line 261 of file `ip6_id.c`.

References `randomid()`, and `randomtab_20`.

Referenced by `in6_pcbconnect()`.

Here is the call graph for this function:



7.38.2.3 u_int32_t ip6_randomid (void)

Definition at line 254 of file `ip6_id.c`.

References `randomid()`, and `randomtab_32`.

Referenced by `ip6_output()`.

Here is the call graph for this function:



7.38.2.4 `static u_int32_t pmod (u_int32_t, u_int32_t, u_int32_t)` [static]

Definition at line 157 of file `ip6_id.c`.

Referenced by `initid()`, and `randomid()`.

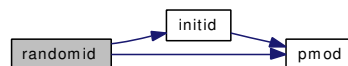
7.38.2.5 `static u_int32_t randomid (struct randomtab *)` [static]

Definition at line 227 of file `ip6_id.c`.

References `initid()`, `pmod()`, `randomtab::ru_a`, `randomtab::ru_b`, `randomtab::ru_counter`, `randomtab::ru_g`, `randomtab::ru_m`, `randomtab::ru_max`, `randomtab::ru_msb`, `randomtab::ru_n`, `randomtab::ru_reseed`, `randomtab::ru_seed`, `randomtab::ru_seed2`, and `randomtab::ru_x`.

Referenced by `ip6_randomflowlabel()`, and `ip6_randomid()`.

Here is the call graph for this function:



7.38.3 Variable Documentation

7.38.3.1 `struct randomtab randomtab_20` [static]

Initial value:

```

{
    20,
    180,
    200000,
    2,
    524269,
    7,
    279936,
    { 2, 3, 14563, 0 },
}

```

Definition at line 136 of file `ip6_id.c`.

Referenced by `ip6_randomflowlabel()`.

7.38.3.2 struct `randomtab_randomtab_32` [static]**Initial value:**

```
{
    32,
    180,
    1000000000,
    2,
    2147483629,
    7,
    1836660096,
    { 2, 3, 59652323, 0 },
}
```

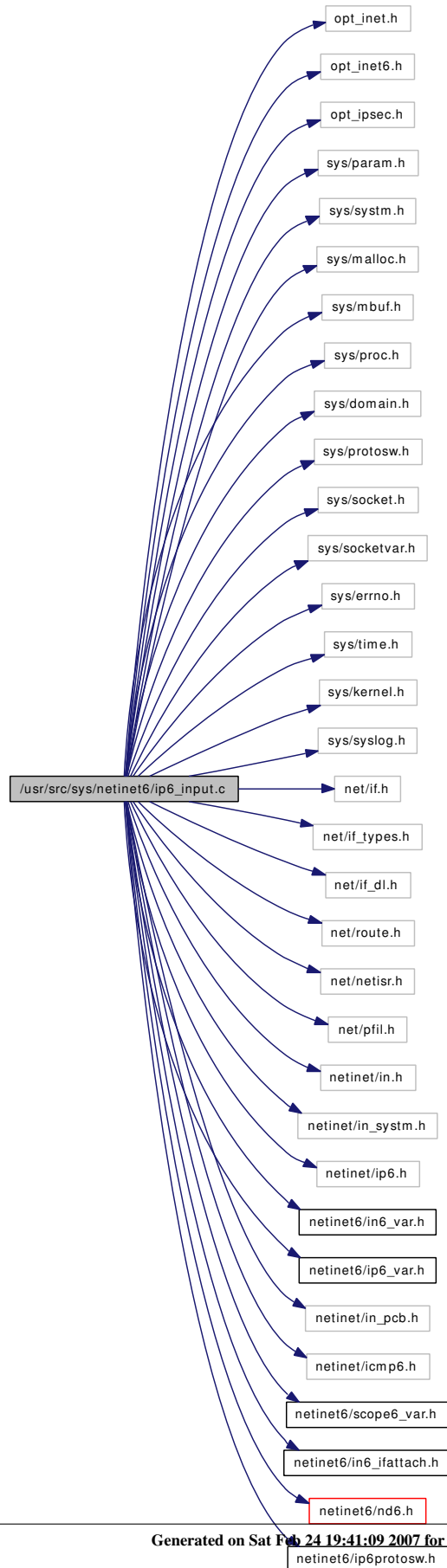
Definition at line 125 of file `ip6_id.c`.

Referenced by `ip6_randomid()`.

7.39 /usr/src/sys/netinet6/ip6_input.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/proc.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/if_dl.h>
#include <net/route.h>
#include <net/netisr.h>
#include <net/pfil.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip6.h>
#include <netinet6/in6_var.h>
#include <netinet6/ip6_var.h>
#include <netinet/in_pcb.h>
#include <netinet/icmp6.h>
#include <netinet6/scope6_var.h>
#include <netinet6/in6_ifattach.h>
#include <netinet6/nd6.h>
#include <netinet6/ip6protosw.h>
```

Include dependency graph for ip6_input.c:



Defines

- #define `M2MMAX` (sizeof(ip6stat.ip6s_m2m)/sizeof(ip6stat.ip6s_m2m[0]))
- #define `rt6_key(r)` ((struct sockaddr_in6 *)((r) → rt_nodes → rn_key))
- #define `IS2292(x, y)` ((in6p → in6p_flags & IN6P_RFC2292) ? (x) : (y))

Functions

- static void `ip6_init2` `__P` ((void *))
- static struct `ip6aux` *`ip6_setdstifaddr` `__P` ((struct mbuf *, struct `in6_ifaddr` *))
- static int `ip6_hopopts_input` `__P` ((u_int32_t *, u_int32_t *, struct mbuf **, int *))
- void `ip6_init` ()
- static void `ip6_init2` (void *dummy)
- `SYSINIT` (netinet6init2, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ip6_init2, NULL)
- void `ip6_input` (struct mbuf *m)
- static struct `ip6aux` * `ip6_setdstifaddr` (struct mbuf *m, struct `in6_ifaddr` *ia6)
- `in6_ifaddr` * `ip6_getdstifaddr` (struct mbuf *m)
- static int `ip6_hopopts_input` (u_int32_t *plenp, u_int32_t *rtalrtp, struct mbuf **mp, int *offp)
- int `ip6_process_hopopts` (struct mbuf *m, u_int8_t *opthead, int hbhlen, u_int32_t *rtalrtp, u_int32_t *plenp)
- int `ip6_unknown_opt` (u_int8_t *optp, struct mbuf *m, int off)
- void `ip6_savecontrol` (struct inpcb *in6p, struct mbuf *m, struct mbuf **mp)
- void `ip6_notify_pmtu` (struct inpcb *in6p, struct `sockaddr_in6` *dst, u_int32_t *mtu)
- char * `ip6_get_prevhdr` (struct mbuf *m, int off)
- int `ip6nexthdr` (struct mbuf *m, int off, int proto, int *nxtp)
- int `ip6lasthdr` (struct mbuf *m, int off, int proto, int *nxtp)
- `ip6aux` * `ip6_addaux` (struct mbuf *m)
- `ip6aux` * `ip6_findaux` (struct mbuf *m)
- void `ip6_delaux` (struct mbuf *m)

Variables

- domain `inet6domain`
- u_char `ip6_protox` [IPPROTO_MAX]
- static struct ifqueue `ip6intrq`
- static int `ip6qmaxlen` = IFQ_MAXLEN
- `in6_ifaddr` * `in6_ifaddr`
- callout `in6_tmpaddrtimer_ch`
- int `ip6_forward_srcrt`
- int `ip6_sourcecheck`
- int `ip6_sourcecheck_interval`
- int `ip6_ours_check_algorithm`
- pfil_head `inet6_pfil_hook`
- `ip6stat` `ip6stat`
- route_in6 `ip6_forward_rt`
- u_char `inet6ctlerrmap` [PRC_NCMDS]

7.39.1 Define Documentation

7.39.1.1 `#define IS2292(x, y) ((in6p → in6p_flags & IN6P_RFC2292) ? (x) : (y))`

Referenced by `ip6_savecontrol()`.

7.39.1.2 `#define M2MMAX (sizeof(ip6stat.ip6s_m2m)/sizeof(ip6stat.ip6s_m2m[0]))`

Referenced by `ip6_input()`.

7.39.1.3 `#define rt6_key(r) ((struct sockaddr_in6 *)((r) → rt_nodes → rn_key))`

Referenced by `ip6_input()`.

7.39.2 Function Documentation

7.39.2.1 `static int ip6_hopopts_input __P ((u_int32_t *, u_int32_t *, struct mbuf **, int *))`
[static]

7.39.2.2 `static struct ip6aux* ip6_setdstifaddr __P ((struct mbuf *, struct in6_ifaddr *))`
[static]

7.39.2.3 `static void ip6_init2 __P ((void *))` [static]

7.39.2.4 `struct ip6aux* ip6_addaux (struct mbuf * m)`

Definition at line 1538 of file `ip6_input.c`.

Referenced by `ip6_setdstifaddr()`.

7.39.2.5 `void ip6_delaux (struct mbuf * m)`

Definition at line 1566 of file `ip6_input.c`.

Referenced by `ip6_input()`.

7.39.2.6 `struct ip6aux* ip6_findaux (struct mbuf * m)`

Definition at line 1556 of file `ip6_input.c`.

Referenced by `ip6_getdstifaddr()`, and `route6_input()`.

7.39.2.7 `char* ip6_get_prevhdr (struct mbuf * m, int off)`

Definition at line 1390 of file `ip6_input.c`.

Referenced by `frag6_input()`, and `rip6_input()`.

7.39.2.8 struct in6_ifaddr* ip6_getdstifaddr (struct mbuf * m)

Definition at line 791 of file ip6_input.c.

References ip6_findaux(), and ip6aux::ip6a_dstia6.

Referenced by frag6_input(), icmp6_reflect(), ip6_input(), ip6_rthdr0(), and ni6_input().

Here is the call graph for this function:

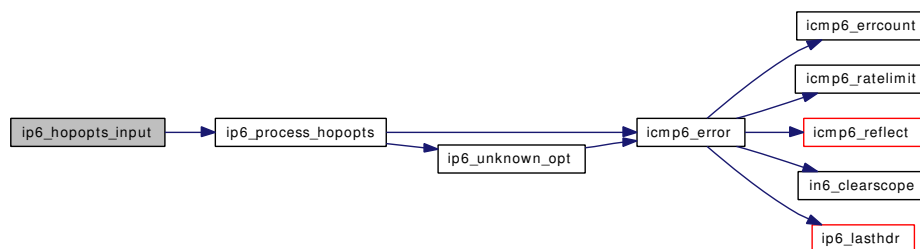
**7.39.2.9 static int ip6_hopopts_input (u_int32_t * plenp, u_int32_t * rtalertp, struct mbuf ** mp, int * offp) [static]**

Definition at line 808 of file ip6_input.c.

References ip6_process_hopopts(), ip6stat::ip6s_tooshort, and ip6stat.

Referenced by ip6_input().

Here is the call graph for this function:

**7.39.2.10 void ip6_init ()**

Definition at line 150 of file ip6_input.c.

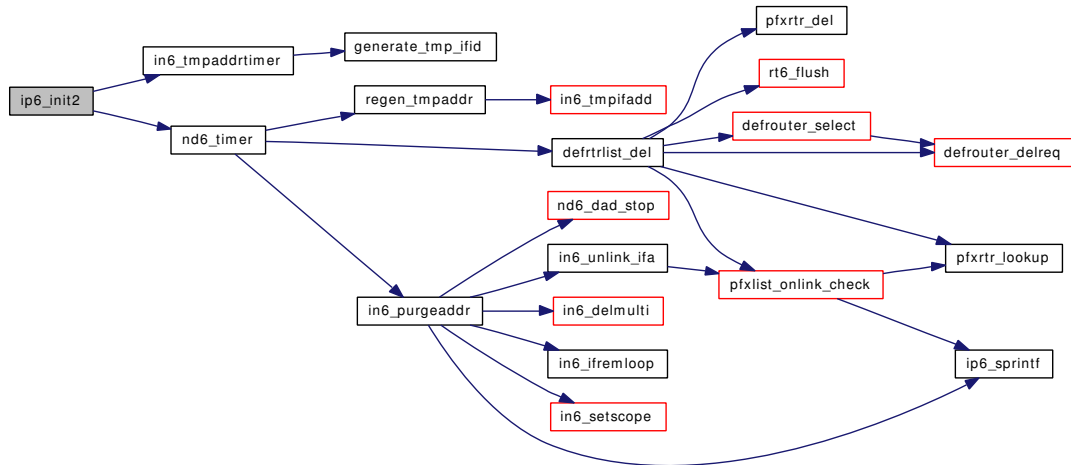
References inet6domain, inet6sw, ip6_protox, ip6protosw::pr_domain, and ip6protosw::pr_protocol.

7.39.2.11 static void ip6_init2 (void * dummy) [static]

Definition at line 197 of file ip6_input.c.

References in6_tmpaddrtimer(), in6_tmpaddrtimer_ch, ip6_desync_factor, ip6_temp_preferred_lifetime, ip6_temp_regen_advance, nd6_timer(), and nd6_timer_ch.

Here is the call graph for this function:

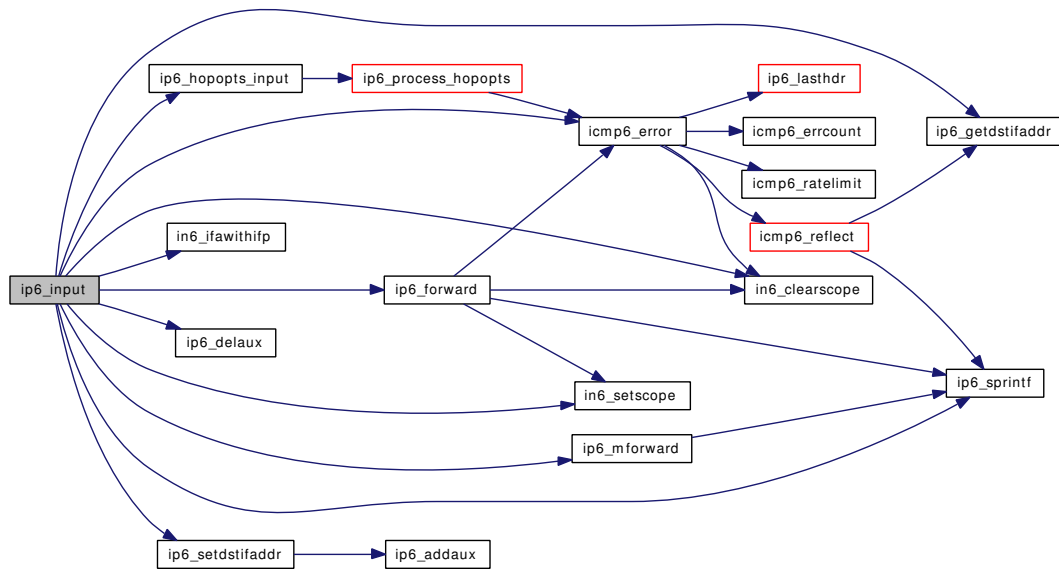


7.39.2.12 void ip6_input (struct mbuf * m)

Definition at line 220 of file ip6_input.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia_ifa, icmp6_error(), IN6_ARE_ADDR_EQUAL, in6_clearscope(), in6_ifaddr, in6_ifawithifp(), IN6_IFF_NOTREADY, in6_ifstat_inc, IN6_IS_ADDR_MC_INTERFACELOCAL, IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, IN6_IS_ADDR_V4COMPAT, IN6_IS_ADDR_V4MAPPED, IN6_LOOKUP_MULTI, in6_setscope(), ipsecstat::in_polvio, INET6_ADDRSTRLEN, inet6_pfil_hook, inet6sw, ip6_delaux(), ip6_forward(), ip6_forwarding, ip6_getdstifaddr(), ip6_hdrnestlimit, ip6_hopopts_input(), ip6_keepfaith, ip6_mforward(), ip6_mroutel, ip6_protox, ip6_setdstifaddr(), ip6_sprintf(), ip6stat::ip6s_badoptions, ip6stat::ip6s_badscope, ip6stat::ip6s_badvers, ip6stat::ip6s_cantforward, ip6stat::ip6s_delivered, ip6stat::ip6s_forward_cachehit, ip6stat::ip6s_forward_cachemiss, ip6stat::ip6s_m1, ip6stat::ip6s_m2m, ip6stat::ip6s_mext1, ip6stat::ip6s_mext2m, ip6stat::ip6s_notmember, ip6stat::ip6s_nxtlist, ip6stat::ip6s_toomanyhdr, ip6stat::ip6s_tooshort, ip6stat::ip6s_toosmall, ip6stat::ip6s_total, ip6stat, ipsec6stat, M2MMAX, M_AUTHIPHDR, M_AUTHIPHDR, M_LOOP, ND6_IFF_IFDISABLED, nd6log, ND_IFINFO, rt6_key, sockaddr_in6::sin6_addr, sockaddr_in6::sin6_family, and sockaddr_in6::sin6_len.

Here is the call graph for this function:



7.39.2.13 int ip6_lasthdr (struct mbuf * m, int off, int proto, int * nxtp)

Definition at line 1510 of file ip6_input.c.

References ip6_nexthdr().

Referenced by icmp6_error().

Here is the call graph for this function:



7.39.2.14 int ip6_nexthdr (struct mbuf * m, int off, int proto, int * nxtp)

Definition at line 1431 of file ip6_input.c.

Referenced by ip6_lasthdr().

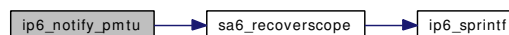
7.39.2.15 void ip6_notify_pmtu (struct inpcb * in6p, struct sockaddr_in6 * dst, u_int32_t * mtu)

Definition at line 1282 of file ip6_input.c.

References IPV6_PATHMTU, and sa6_recoverscope().

Referenced by in6_pcbnotify().

Here is the call graph for this function:



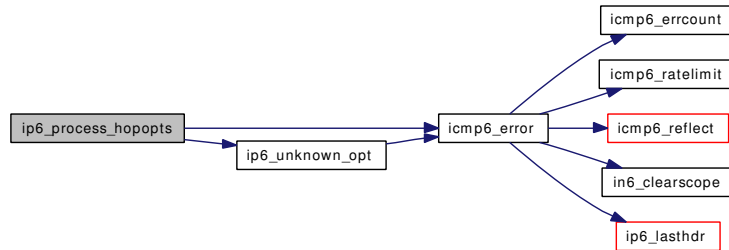
7.39.2.16 `int ip6_process_hopopts (struct mbuf * m, u_int8_t * opthead, int hbhlen, u_int32_t * rtalertp, u_int32_t * plenp)`

Definition at line 866 of file ip6_input.c.

References icmp6_error(), ip6_unknown_opt(), ip6stat::ip6s_badoptions, ip6stat::ip6s_toosmall, and ip6stat.

Referenced by ip6_hopopts_input(), and ip6_output().

Here is the call graph for this function:



7.39.2.17 `void ip6_savecontrol (struct inpcb * in6p, struct mbuf * m, struct mbuf ** mp)`

Definition at line 1047 of file ip6_input.c.

References elen, in6_clearscope(), ip6stat::ip6s_tooshort, ip6stat, in6_pktinfo::ipi6_addr, in6_pktinfo::ipi6_ifindex, IPV6_2292DSTOPTS, IPV6_2292HOPLIMIT, IPV6_2292HOPOPTS, IPV6_2292PKTINFO, IPV6_2292RTHDR, IPV6_DSTOPTS, IPV6_HOPLIMIT, IPV6_HOPOPTS, IPV6_PKTINFO, IPV6_RTHDR, IPV6_TCLASS, and IS2292.

Referenced by icmp6_rip6_input(), rip6_input(), and udp6_append().

Here is the call graph for this function:



7.39.2.18 `static struct ip6aux* ip6_setdstifaddr (struct mbuf * m, struct in6_ifaddr * ia6) [static]`

Definition at line 778 of file ip6_input.c.

References ip6_addaux(), and ip6aux::ip6a_dstia6.

Referenced by ip6_input().

Here is the call graph for this function:



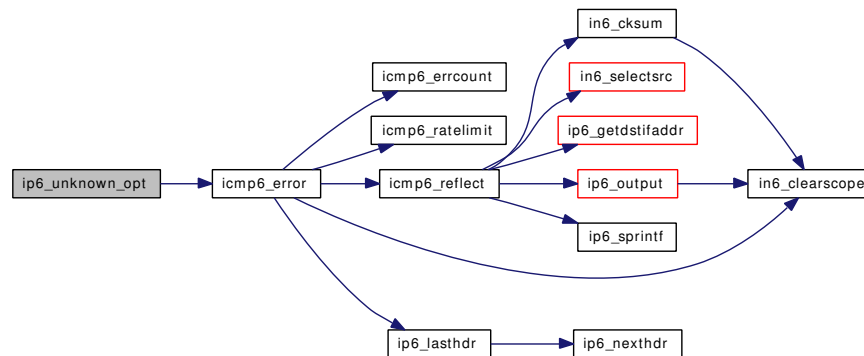
7.39.2.19 `int ip6_unknown_opt (u_int8_t * optp, struct mbuf * m, int off)`

Definition at line 1003 of file ip6_input.c.

References icmp6_error(), IN6_IS_ADDR_MULTICAST, ip6stat::ip6s_badoptions, and ip6stat.

Referenced by dest6_input(), and ip6_process_hopopts().

Here is the call graph for this function:

**7.39.2.20** `SYSINIT (netinet6init2, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ip6_init2, NULL)`**7.39.3 Variable Documentation****7.39.3.1** `struct in6_ifaddr* in6_ifaddr`

Definition at line 124 of file ip6_input.c.

Referenced by ip6_input().

7.39.3.2 `struct callout in6_tmpaddrtimer_ch`

Definition at line 69 of file in6_ifattach.c.

7.39.3.3 `struct pfil_head inet6_pfil_hook`

Definition at line 134 of file ip6_input.c.

Referenced by ip6_forward(), ip6_input(), and ip6_output().

7.39.3.4 `u_char inet6ctlerrmap[PRC_NCMDS]`

Initial value:

```

{
    0,          0,          0,          0,
    0,          EMSGSIZE,  EHOSTDOWN,  EHOSTUNREACH,
    EHOSTUNREACH, EHOSTUNREACH, ECONNREFUSED, ECONNREFUSED,
    EMSGSIZE,  EHOSTUNREACH, 0,          0,

```

```
    0,          0,          0,          0,  
    ENOPROTOPT  
}
```

Definition at line 1580 of file ip6_input.c.

Referenced by in6_pcbnotify(), rip6_ctlinput(), sctp6_ctlinput(), and udp6_ctlinput().

7.39.3.5 struct domain inet6domain

Definition at line 369 of file in6_proto.c.

7.39.3.6 struct route_in6 ip6_forward_rt

Definition at line 85 of file ip6_forward.c.

Referenced by frag6_slowtimo(), and ip6_forward().

7.39.3.7 int ip6_forward_srcrt

Definition at line 128 of file ip6_input.c.

7.39.3.8 int ip6_ours_check_algorithm

Definition at line 132 of file ip6_input.c.

7.39.3.9 u_char ip6_protox[IPPROTO_MAX]

Definition at line 121 of file ip6_input.c.

Referenced by icmp6_notify_error(), ip6_init(), and ip6_input().

7.39.3.10 int ip6_sourcecheck

Definition at line 129 of file ip6_input.c.

7.39.3.11 int ip6_sourcecheck_interval

Definition at line 130 of file ip6_input.c.

7.39.3.12 struct ifqueue ip6intrq [static]

Definition at line 122 of file ip6_input.c.

7.39.3.13 int ip6qmaxlen = IFQ_MAXLEN [static]

Definition at line 123 of file ip6_input.c.

7.39.3.14 struct ip6stat ip6stat

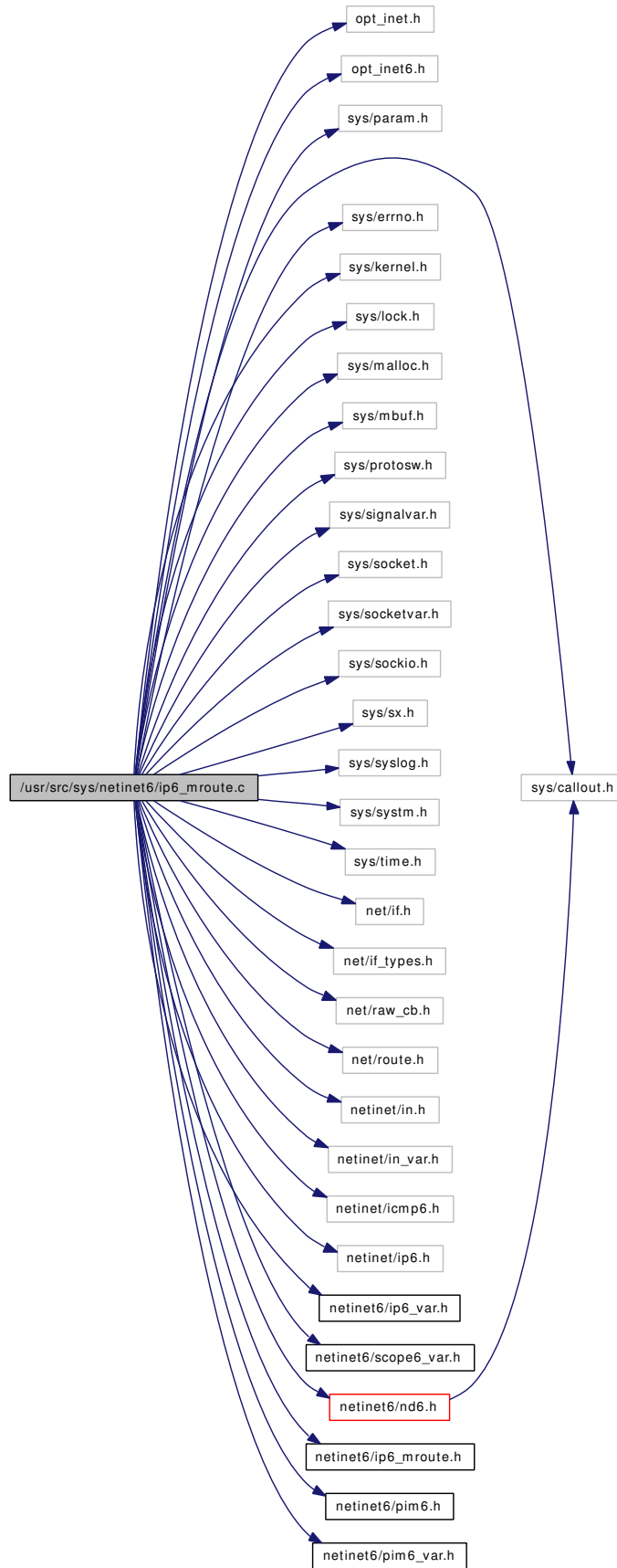
Definition at line 136 of file ip6_input.c.

Referenced by ip6_hopopts_input(), ip6_input(), ip6_process_hopopts(), ip6_savecontrol(), and ip6_unknown_opt().

7.40 /usr/src/sys/netinet6/ip6_mroute.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/callout.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/protosw.h>
#include <sys/signalvar.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sockio.h>
#include <sys/sx.h>
#include <sys/syslog.h>
#include <sys/system.h>
#include <sys/time.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/raw_cb.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/icmp6.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet6/nd6.h>
#include <netinet6/ip6_mroute.h>
#include <netinet6/pim6.h>
#include <netinet6/pim6_var.h>
```

Include dependency graph for ip6_mroute.c:



Defines

- #define `M_HASCL(m)` $((m) \rightarrow m_flags \& M_EXT)$
- #define `NO_RTE_FOUND` 0x1
- #define `RTE_FOUND` 0x2
- #define `EXPIRE_TIMEOUT` (hz / 4)
- #define `UPCALL_EXPIRE` 6
- #define `ENCAP_HOPS` 64
- #define `MF6CHASH(a, g)`
- #define `MF6CFIND(o, g, rt)`
- #define `TV_DELTA(a, b, delta)`
- #define `TV_LT(a, b)`
- #define `MC6_SEND(ip6, mifp, m)`
- #define `PIM6_CHECKSUM`

Functions

- static `MALLOC_DEFINE` (M_MRTABLE6, "mf6c", "multicast forwarding cache entry")
- static int `ip6_mdq __P` ((struct mbuf *, struct ifnet *, struct `mf6c` *)
- static void `phyint_send __P` ((struct ip6_hdr *, struct `mif6` *, struct mbuf *)
- static int `set_pim6 __P` ((int *)
- static int `socket_send __P` ((struct socket *, struct mbuf *, struct `sockaddr_in6` *)
- static void `expire_upcalls __P` ((void *)
- static int `get_sg_cnt __P` ((struct `sioc_sg_req6` *)
- static int `get_mif6_cnt __P` ((struct `sioc_mif_req6` *)
- static int `ip6_mrouterr_init __P` ((struct socket *, int, int)
- static int `add_m6if __P` ((struct `mif6ctl` *)
- static int `del_m6if __P` ((`mifi_t` *)
- static int `add_m6fc __P` ((struct `mf6cctl` *)
- int `ip6_mrouterr_set` (struct socket *so, struct sockopt *sopt)
- int `ip6_mrouterr_get` (struct socket *so, struct sockopt *sopt)
- int `mrt6_ioctl` (int cmd, caddr_t data)
- static int `get_sg_cnt` (struct `sioc_sg_req6` *req)
- static int `get_mif6_cnt` (struct `sioc_mif_req6` *req)
- static int `set_pim6` (int *i)
- static int `ip6_mrouterr_init` (struct socket *so, int v, int cmd)
- int `ip6_mrouterr_done` ()
- static int `add_m6if` (struct `mif6ctl` *mifcp)
- static int `del_m6if` (`mifi_t` *mifip)
- static int `add_m6fc` (struct `mf6cctl` *mfccp)
- static int `del_m6fc` (struct `mf6cctl` *mfccp)
- static int `socket_send` (struct socket *s, struct mbuf *mm, struct `sockaddr_in6` *src)
- int `ip6_mforward` (struct ip6_hdr *ip6, struct ifnet *ifp, struct mbuf *m)
- static void `expire_upcalls` (void *unused)
- static int `ip6_mdq` (struct mbuf *m, struct ifnet *ifp, struct `mf6c` *rt)
- static void `phyint_send` (struct ip6_hdr *ip6, struct `mif6` *mifp, struct mbuf *m)
- static int `register_send` (struct ip6_hdr *ip6, struct `mif6` *mif, struct mbuf *m)
- int `pim6_input` (struct mbuf **mp, int *offp, int proto)

Variables

- socket * `ip6_mrouter` = NULL
- int `ip6_mrouter_ver` = 0
- int `ip6_mrtproto` = IPPROTO_PIM
- `mrt6stat` `mrt6stat`
- `mf6c` * `mf6ctable` [MF6CTBLSIZ]
- u_char `n6expire` [MF6CTBLSIZ]
- static struct `mif6` `mif6table` [MAXMIFS]
- static struct ifnet * `multicast_register_if6`
- static `mifi_t` `nummifs` = 0
- static `mifi_t` `reg_mif_num` = (mifi_t)-1
- static struct `pim6stat` `pim6stat`
- static int `pim6`
- static struct callout `expire_upcalls_ch`
- static struct `sockaddr_in6` `sin6` = { sizeof(sin6), AF_INET6 }

7.40.1 Define Documentation

7.40.1.1 #define ENCAP_HOPS 64

Definition at line 181 of file ip6_mroute.c.

7.40.1.2 #define EXPIRE_TIMEOUT (hz / 4)

Definition at line 159 of file ip6_mroute.c.

Referenced by ip6_mrouter_init().

7.40.1.3 #define M_HASCL(m) ((m) → m_flags & M_EXT)

Definition at line 122 of file ip6_mroute.c.

Referenced by ip6_mdq(), ip6_mforward(), and phyint_send().

7.40.1.4 #define MC6_SEND(ip6, mifp, m)

Value:

```
do {
    if ((mifp)->m6_flags & MIFF_REGISTER)
        register_send(ip6, (mifp), (m));
    else
        phyint_send(ip6, (mifp), (m));
} while (/*CONSTCOND*/ 0)
```

Referenced by ip6_mdq().

7.40.1.5 #define MF6CFIND(o, g, rt)**Value:**

```

do { \
    struct mf6c *_rt = mf6ctable[MF6CHASH(o,g)]; \
    rt = NULL; \
    mrt6stat.mrt6s_mfc_lookups++; \
    while (_rt) { \
        if (IN6_ARE_ADDR_EQUAL(&_rt->mf6c_origin.sin6_addr, &(o)) && \
            IN6_ARE_ADDR_EQUAL(&_rt->mf6c_mcastgrp.sin6_addr, &(g)) && \
            (_rt->mf6c_stall == NULL)) { \
            rt = _rt; \
            break; \
        } \
        _rt = _rt->mf6c_next; \
    } \
    if (rt == NULL) { \
        mrt6stat.mrt6s_mfc_misses++; \
    } \
} while (/*CONSTCOND*/ 0)

```

Definition at line 205 of file ip6_mroute.c.

Referenced by add_m6fc(), get_sg_cnt(), and ip6_mforward().

7.40.1.6 #define MF6CHASH(a, g)**Value:**

```

MF6CHASHMOD((a).s6_addr32[0] ^ (a).s6_addr32[1] ^ \
            (a).s6_addr32[2] ^ (a).s6_addr32[3] ^ \
            (g).s6_addr32[0] ^ (g).s6_addr32[1] ^ \
            (g).s6_addr32[2] ^ (g).s6_addr32[3])

```

Definition at line 195 of file ip6_mroute.c.

Referenced by add_m6fc(), del_m6fc(), and ip6_mforward().

7.40.1.7 #define NO_RTE_FOUND 0x1

Definition at line 142 of file ip6_mroute.c.

7.40.1.8 #define PIM6_CHECKSUM**7.40.1.9 #define RTE_FOUND 0x2**

Definition at line 143 of file ip6_mroute.c.

7.40.1.10 #define TV_DELTA(a, b, delta)**Value:**

```

do { \
    int xxs; \

```

```
        \
        delta = (a).tv_usec - (b).tv_usec; \
        if ((xxs = (a).tv_sec - (b).tv_sec)) { \
            switch (xxs) { \
                case 2: \
                    delta += 1000000; \
                    /* FALLTHROUGH */ \
                case 1: \
                    delta += 1000000; \
                    break; \
                default: \
                    delta += (1000000 * xxs); \
            } \
        } \
    } while (/*CONSTCOND*/ 0)
```

Definition at line 227 of file ip6_mroute.c.

7.40.1.11 #define TV_LT(a, b)

Value:

```
((a).tv_usec < (b).tv_usec && \
 (a).tv_sec <= (b).tv_sec) || (a).tv_sec < (b).tv_sec)
```

Definition at line 245 of file ip6_mroute.c.

7.40.1.12 #define UPCALL_EXPIRE 6

Definition at line 160 of file ip6_mroute.c.

Referenced by ip6_mforward().

7.40.2 Function Documentation

7.40.2.1 `static int del_m6fc __P ((struct mf6ctl *))` [static]

7.40.2.2 `static int del_m6if __P ((mifi_t *))` [static]

7.40.2.3 `static int add_m6if __P ((struct mif6ctl *))` [static]

7.40.2.4 `static int ip6_mrouter_init __P ((struct socket *, int, int))` [static]

7.40.2.5 `static int get_mif6_cnt __P ((struct sioc_mif_req6 *))` [static]

7.40.2.6 `static int get_sg_cnt __P ((struct sioc_sg_req6 *))` [static]

7.40.2.7 `static void expire_upcalls __P ((void *))` [static]

7.40.2.8 `static int socket_send __P ((struct socket *, struct mbuf *, struct sockaddr_in6 *))`
[static]

7.40.2.9 `static int set_pim6 __P ((int *))` [static]

7.40.2.10 `static int register_send __P ((struct ip6_hdr *, struct mif6 *, struct mbuf *))` [static]

7.40.2.11 `static int ip6_mdq __P ((struct mbuf *, struct ifnet *, struct mf6c *))` [static]

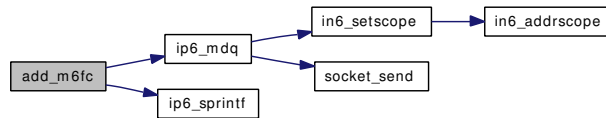
7.40.2.12 `static int add_m6fc (struct mf6ctl * mf6cp)` [static]

Definition at line 720 of file `ip6_mroute.c`.

References `rtdetq::ifp`, `IN6_ARE_ADDR_EQUAL`, `INET6_ADDRSTRLEN`, `ip6_mdq()`, `ip6_sprintf()`, `rtdetq::m`, `mf6c::mf6c_byte_cnt`, `mf6c::mf6c_expire`, `mf6c::mf6c_ifset`, `mf6c::mf6c_mcastgrp`, `mf6c::mf6c_next`, `mf6c::mf6c_origin`, `mf6c::mf6c_parent`, `mf6c::mf6c_pkt_cnt`, `mf6c::mf6c_stall`, `mf6c::mf6c_wrong_if`, `MF6CFIND`, `MF6CHASH`, `rtdetq::next`, and `sockaddr_in6::sin6_addr`.

Referenced by `ip6_mrouter_set()`.

Here is the call graph for this function:



7.40.2.13 `static int add_m6if (struct mif6ctl * mif6cp)` [static]

Definition at line 577 of file `ip6_mroute.c`.

References `mif6::m6_bytes_in`, `mif6::m6_bytes_out`, `mif6::m6_flags`, `mif6::m6_ifp`, `mif6::m6_pkt_in`, `mif6::m6_pkt_out`, `mif6::m6_rate_limit`, `MAXMIFS`, `mif6table`, and `MIFF_REGISTER`.

Referenced by `ip6_mrouter_set()`.

7.40.2.14 static int del_m6fc (struct mf6ctl * mfccp) [static]

Definition at line 909 of file ip6_mroute.c.

References IN6_ARE_ADDR_EQUAL, INET6_ADDRSTRLEN, ip6_sprintf(), mf6c::mf6c_mcastgrp, mf6c::mf6c_next, mf6c::mf6c_origin, mf6c::mf6c_stall, MF6CHASH, and sockaddr_in6::sin6_addr.

Referenced by ip6_mrouter_set().

Here is the call graph for this function:

**7.40.2.15** static int del_m6if (mifi_t * mifip) [static]

Definition at line 661 of file ip6_mroute.c.

References mif6::m6_flags, mif6::m6_ifp, mif6table, and MIFF_REGISTER.

Referenced by ip6_mrouter_set().

7.40.2.16 static void expire_upcalls (void * unused) [static]

Definition at line 1264 of file ip6_mroute.c.

References INET6_ADDRSTRLEN, ip6_sprintf(), rtdetq::m, mf6c::mf6c_expire, mf6c::mf6c_mcastgrp, mf6c::mf6c_next, mf6c::mf6c_origin, mf6c::mf6c_stall, MF6CTBLSIZ, mrt6stat::mrt6s_cache_cleanups, mrt6stat, rtdetq::next, and sockaddr_in6::sin6_addr.

Referenced by ip6_mrouter_init().

Here is the call graph for this function:

**7.40.2.17** static int get_mif6_cnt (struct sioc_mif_req6 * req) [static]

Definition at line 403 of file ip6_mroute.c.

References mif6::m6_bytes_in, mif6::m6_bytes_out, mif6::m6_pkt_in, mif6::m6_pkt_out, and mif6table.

Referenced by mrt6_ioctl().

7.40.2.18 static int get_sg_cnt (struct sioc_sg_req6 * req) [static]

Definition at line 377 of file ip6_mroute.c.

References mf6c::mf6c_byte_cnt, mf6c::mf6c_pkt_cnt, mf6c::mf6c_wrong_if, and MF6CFIND.

Referenced by mrt6_ioctl().

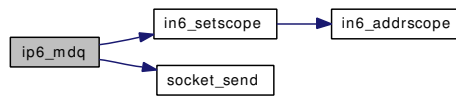
7.40.2.19 static int ip6_mdq (struct mbuf * m, struct ifnet * ifp, struct mf6c * rt) [static]

Definition at line 1325 of file ip6_mroute.c.

References IF_ISSET, mrt6msg::im6_mbz, omrt6msg::im6_mbz, mrt6msg::im6_mif, omrt6msg::im6_mif, mrt6msg::im6_msgtype, omrt6msg::im6_msgtype, in6_setscope(), mif6::m6_bytes_in, mif6::m6_bytes_out, mif6::m6_flags, mif6::m6_ifp, mif6::m6_pkt_in, mif6::m6_pkt_out, M_HASCL, M_LOOP, MC6_SEND, mf6c::mf6c_byte_cnt, mf6c::mf6c_ifset, mf6c::mf6c_parent, mf6c::mf6c_pkt_cnt, mf6c::mf6c_wrong_if, mif6table, MIFF_REGISTER, MRT6_INIT, MRT6_OINIT, MRT6MSG_WRONGMIF, mrt6stat::mrt6s_upcalls, mrt6stat::mrt6s_upq_sockfull, mrt6stat::mrt6s_wrong_if, mrt6stat, sin6, sockaddr_in6::sin6_addr, and socket_send().

Referenced by add_m6fc(), and ip6_mforward().

Here is the call graph for this function:



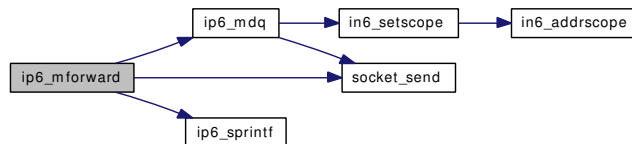
7.40.2.20 int ip6_mforward (struct ip6_hdr * ip6, struct ifnet * ifp, struct mbuf * m)

Definition at line 996 of file ip6_mroute.c.

References GET_TIME, omrt6msg::im6_mbz, omrt6msg::im6_mif, omrt6msg::im6_msgtype, IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_MC_INTFACELocal, IN6_IS_ADDR_MC_LINKLOCAL, IN6_IS_ADDR_UNSPECIFIED, INET6_ADDRSTRLEN, ip6_log_interval, ip6_log_time, ip6_mdq(), ip6_sprintf(), rtdetq::m, mif6::m6_ifp, M_HASCL, MAX_UPQ6, MF6C_INCOMPLETE_PARENT, mf6c::mf6c_mcastgrp, mf6c::mf6c_next, mf6c::mf6c_origin, mf6c::mf6c_stall, MF6CFIND, MF6CHASH, mif6table, MRT6_INIT, MRT6_OINIT, MRT6MSG_NOCACHE, mrt6stat::mrt6s_no_route, mrt6stat::mrt6s_upcalls, mrt6stat::mrt6s_upq_ovflw, mrt6stat::mrt6s_upq_sockfull, mrt6stat, rtdetq::next, sin6, sockaddr_in6::sin6_addr, socket_send(), and UPCALL_EXPIRE.

Referenced by ip6_input(), and ip6_output().

Here is the call graph for this function:



7.40.2.21 int ip6_mrouter_done ()

Definition at line 481 of file ip6_mroute.c.

References rtdetq::m, mf6ctable, MF6CTBSIZ, mif6table, MIFF_REGISTER, and rtdetq::next.

Referenced by ip6_mrouter_set(), and rip6_detach().

7.40.2.22 int ip6_mrouter_get (struct socket * so, struct sockopt * sopt)

Definition at line 338 of file ip6_mroute.c.

References MRT6_PIM.

Referenced by rip6_ctloutput().

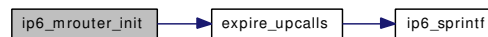
7.40.2.23 static int ip6_mrouter_init (struct socket * so, int v, int cmd) [static]

Definition at line 435 of file ip6_mroute.c.

References EXPIRE_TIMEOUT, expire_upcalls(), and mf6table.

Referenced by ip6_mrouter_set().

Here is the call graph for this function:

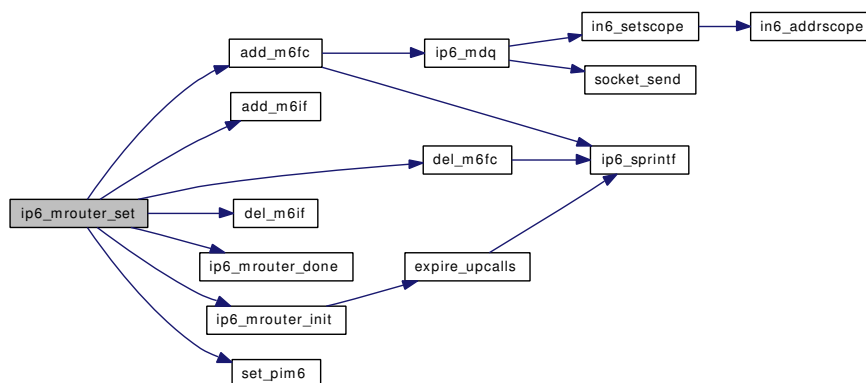
**7.40.2.24 int ip6_mrouter_set (struct socket * so, struct sockopt * sopt)**

Definition at line 268 of file ip6_mroute.c.

References add_m6fc(), add_m6if(), del_m6fc(), del_m6if(), ip6_mrouter_done(), ip6_mrouter_init(), MRT6_ADD_MFC, MRT6_ADD_MIF, MRT6_DEL_MFC, MRT6_DEL_MIF, MRT6_DONE, MRT6_INIT, MRT6_OINIT, MRT6_PIM, and set_pim6().

Referenced by rip6_ctloutput().

Here is the call graph for this function:

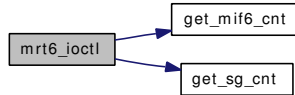
**7.40.2.25 static MALLOC_DEFINE (M_MRTABLE6, "mf6c", "multicast forwarding cache entry") [static]****7.40.2.26 int mrt6_ioctl (int cmd, caddr_t data)**

Definition at line 359 of file ip6_mroute.c.

References `get_mif6_cnt()`, `get_sg_cnt()`, `SIOCGETMIFCNT_IN6`, and `SIOCGETSGCNT_IN6`.

Referenced by `in6_control()`.

Here is the call graph for this function:

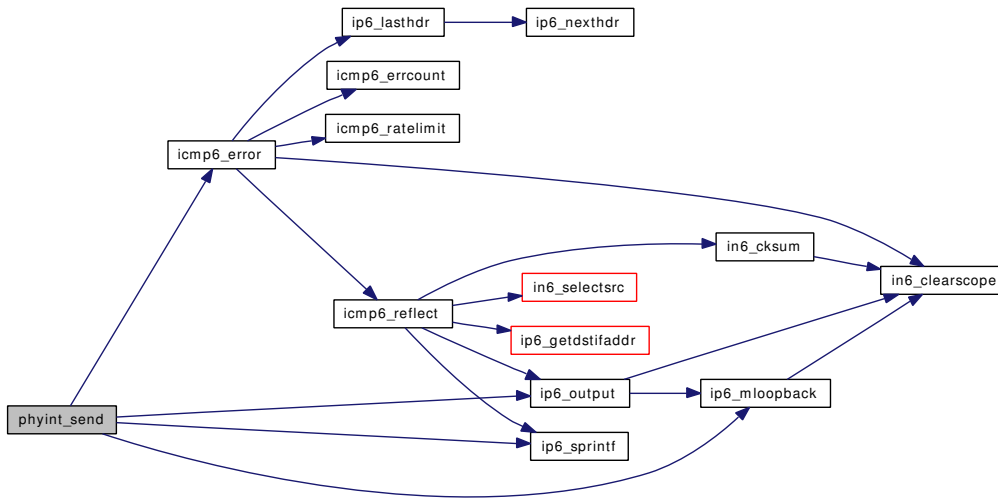


7.40.2.27 static void `phyint_send (struct ip6_hdr * ip6, struct mif6 * mifp, struct mbuf * m)` [static]

Definition at line 1506 of file `ip6_mroute.c`.

References `icmp6_error()`, `ip6_moptions::im6o_multicast_hlim`, `ip6_moptions::im6o_multicast_ifp`, `ip6_moptions::im6o_multicast_loop`, `IN6_LINKMTU`, `IN6_LOOKUP_MULTI`, `INET6_ADDRSTRLEN`, `ip6_mcast_pmtu`, `ip6_mloopback()`, `ip6_output()`, `ip6_sprintf()`, `IPV6_FORWARDING`, `mif6::m6_ifp`, `M_HASCL`, `mif6table`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, and `sockaddr_in6::sin6_len`.

Here is the call graph for this function:

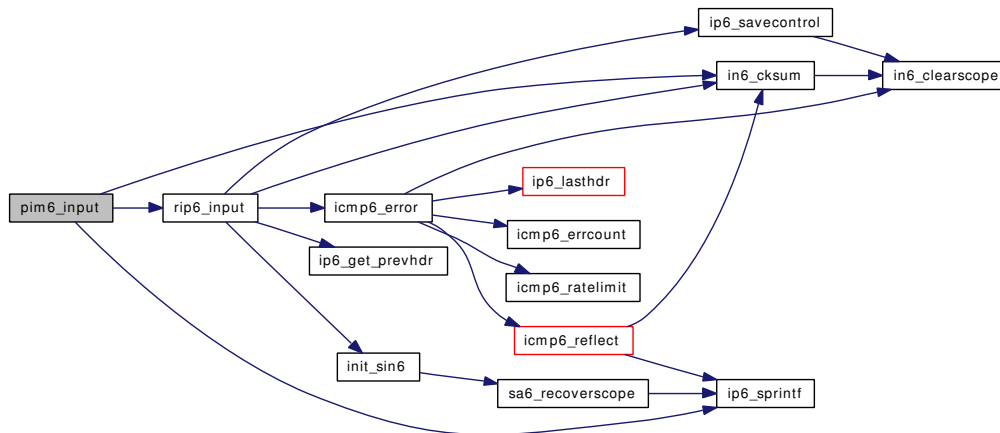


7.40.2.28 int `pim6_input (struct mbuf ** mp, int * offp, int proto)`

Definition at line 1699 of file `ip6_mroute.c`.

References `in6_cksum()`, `IN6_IS_ADDR_MULTICAST`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `mif6table`, `PIM6_REG_MINLEN`, `pim6stat::pim6s_rcv_badregisters`, `pim6stat::pim6s_rcv_badsum`, `pim6stat::pim6s_rcv_badversion`, `pim6stat::pim6s_rcv_registers`, `pim6stat::pim6s_rcv_tooshort`, `pim6stat::pim6s_rcv_total`, `pim6stat`, `PIM_MINLEN`, `PIM_NULL_REGISTER`, `PIM_REGISTER`, `PIM_VERSION`, `rip6_input()`, and `sockaddr_in6::sin6_family`.

Here is the call graph for this function:

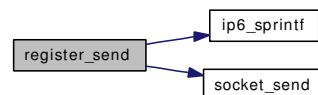


7.40.2.29 static int register_send (struct ip6_hdr * ip6, struct mif6 * mif, struct mbuf * m) [static]

Definition at line 1624 of file ip6_mroute.c.

References mrt6msg::im6_mbz, mrt6msg::im6_mif, mrt6msg::im6_msgtype, INET6_ADDRSTRLEN, ip6_sprintf(), mif6table, MRT6MSG_WHOLEPKT, mrt6stat::mrt6s_upcalls, mrt6stat::mrt6s_upq_sockfull, mrt6stat, pim6stat::pim6s_snd_registers, pim6stat, sin6, sockaddr_in6::sin6_addr, and socket_send().

Here is the call graph for this function:



7.40.2.30 static int set_pim6 (int * i) [static]

Definition at line 420 of file ip6_mroute.c.

Referenced by ip6_mrouter_set().

7.40.2.31 static int socket_send (struct socket * s, struct mbuf * mm, struct sockaddr_in6 * src) [static]

Definition at line 959 of file ip6_mroute.c.

Referenced by ip6_mdq(), ip6_mforward(), and register_send().

7.40.3 Variable Documentation

7.40.3.1 struct callout `expire_upcalls_ch` [static]

Definition at line 262 of file ip6_mroute.c.

7.40.3.2 struct socket* `ip6_mrouter` = NULL

Definition at line 137 of file ip6_mroute.c.

Referenced by ip6_input(), ip6_output(), mld6_sendpkt(), and rip6_detach().

7.40.3.3 int `ip6_mrouter_ver` = 0

Definition at line 138 of file ip6_mroute.c.

7.40.3.4 int `ip6_mrtproto` = IPPROTO_PIM

Definition at line 139 of file ip6_mroute.c.

7.40.3.5 struct mf6c* `mf6ctable`[MF6CTBLSIZ]

Definition at line 145 of file ip6_mroute.c.

Referenced by ip6_mrouter_done(), and ip6_mrouter_init().

7.40.3.6 struct mif6 `mif6table`[MAXMIFS] [static]

Definition at line 147 of file ip6_mroute.c.

Referenced by add_m6if(), del_m6if(), get_mif6_cnt(), ip6_mdq(), ip6_mforward(), ip6_mrouter_done(), phyint_send(), pim6_input(), and register_send().

7.40.3.7 struct mrt6stat `mrt6stat`

Definition at line 140 of file ip6_mroute.c.

Referenced by expire_upcalls(), ip6_mdq(), ip6_mforward(), and register_send().

7.40.3.8 struct ifnet* `multicast_register_if6` [static]

Definition at line 179 of file ip6_mroute.c.

7.40.3.9 u_char `n6expire`[MF6CTBLSIZ]

Definition at line 146 of file ip6_mroute.c.

7.40.3.10 mifi_t `nummifs` = 0 [static]

Definition at line 186 of file ip6_mroute.c.

7.40.3.11 int pim6 [static]

Definition at line 190 of file ip6_mroute.c.

7.40.3.12 struct pim6stat pim6stat [static]

Definition at line 189 of file ip6_mroute.c.

Referenced by pim6_input(), and register_send().

7.40.3.13 mifi_t reg_mif_num = (mifi_t)-1 [static]

Definition at line 187 of file ip6_mroute.c.

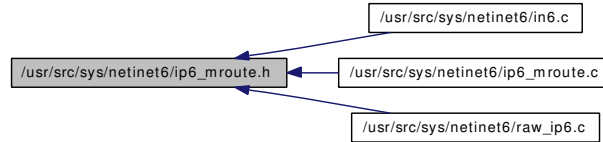
7.40.3.14 struct sockaddr_in6 sin6 = { sizeof(sin6), AF_INET6 } [static]

Definition at line 571 of file ip6_mroute.c.

Referenced by gif_validate6(), icmp6_redirect_input(), icmp6_redirect_output(), icmp6_reflect(), in6_addroute(), in6_ifdetach(), in6_ifinit(), in6_lifaddr_ioctl(), in6_pcbbind(), in6_pcbconnect(), in6_pcbldaddr(), in6_sin6_2_sin(), in6_sin6_2_sin_in_sock(), in6_sin_2_v4mapsin6(), in6_sockaddr(), init_sin6(), ip6_mdq(), ip6_mforward(), nd6_lookup(), nd6_sysctl_prlist(), register_send(), sctp6_connect(), sctp6_getaddr(), sctp6_getpeeraddr(), sctp6_in6getaddr(), sctp6_peeraddr(), sctp6_send(), udp6_output(), and udp6_send().

7.41 /usr/src/sys/netinet6/ip6_mroute.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [if_set](#)
- struct [mif6ctl](#)
- struct [mf6cctl](#)
- struct [mrt6stat](#)
- struct [omrt6msg](#)
- struct [mrt6msg](#)
- struct [sioc_sg_req6](#)
- struct [sioc_mif_req6](#)
- struct [mif6](#)
- struct [mf6c](#)
- struct [rtdetq](#)

Defines

- #define [MRT6_OINIT](#) 100
- #define [MRT6_DONE](#) 101
- #define [MRT6_ADD_MIF](#) 102
- #define [MRT6_DEL_MIF](#) 103
- #define [MRT6_ADD_MFC](#) 104
- #define [MRT6_DEL_MFC](#) 105
- #define [MRT6_PIM](#) 107
- #define [MRT6_INIT](#) 108
- #define [GET_TIME](#)(t) ((t) = time)
- #define [MAXMIFS](#) 64
- #define [IF_SETSIZE](#) 256
- #define [NIFBITS](#) (sizeof(if_mask) * NBBY)
- #define [howmany](#)(x, y) (((x) + ((y) - 1)) / (y))
- #define [IF_SET](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] |= (1 << ((n) % NIFBITS)))
- #define [IF_CLR](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] &= ~(1 << ((n) % NIFBITS)))
- #define [IF_ISSET](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] & (1 << ((n) % NIFBITS)))
- #define [IF_COPY](#)(f, t) bcopy(f, t, sizeof(*(f)))
- #define [IF_ZERO](#)(p) bzero(p, sizeof(*(p)))
- #define [MIFF_REGISTER](#) 0x1
- #define [MRT6MSG_NOCACHE](#) 1
- #define [MRT6MSG_WRONGMIF](#) 2
- #define [MRT6MSG_WHOLEPKT](#) 3
- #define [MF6C_INCOMPLETE_PARENT](#) ((mifi_t)-1)

- #define [MF6CTBSIZ](#) 256
- #define [MF6HASHMOD](#)(h) ((h) & (MF6CTBSIZ - 1))
- #define [MAX_UPQ6](#) 4

Typedefs

- typedef u_short [mifi_t](#)
- typedef u_int32_t [if_mask](#)

Functions

- int [ip6_mrouter_set](#) [__P](#) ((struct socket *so, struct sockopt *sopt))
- int [ip6_mrouter_done](#) [__P](#) ((void))
- int [mrt6_ioctl](#) [__P](#) ((int, caddr_t))

7.41.1 Define Documentation

7.41.1.1 #define [GET_TIME](#)(t) ((t) = time)

Definition at line 69 of file ip6_mroute.h.

Referenced by [ip6_mforward](#)() .

7.41.1.2 #define [howmany](#)(x, y) (((x) + ((y) - 1)) / (y))

Definition at line 86 of file ip6_mroute.h.

7.41.1.3 #define [IF_CLR](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] &= ~(1 << ((n) % NIFBITS)))

Definition at line 94 of file ip6_mroute.h.

7.41.1.4 #define [IF_COPY](#)(f, t) bcopy(f, t, sizeof(*(f)))

Definition at line 96 of file ip6_mroute.h.

7.41.1.5 #define [IF_ISSET](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] & (1 << ((n) % NIFBITS)))

Definition at line 95 of file ip6_mroute.h.

Referenced by [ip6_mdq](#)() .

7.41.1.6 #define [IF_SET](#)(n, p) ((p) → ifs_bits[(n)/NIFBITS] |= (1 << ((n) % NIFBITS)))

Definition at line 93 of file ip6_mroute.h.

7.41.1.7 #define [IF_SETSIZE](#) 256

Definition at line 79 of file ip6_mroute.h.

7.41.1.8 #define IF_ZERO(p) bzero(p, sizeof(*(p)))

Definition at line 97 of file ip6_mroute.h.

7.41.1.9 #define MAX_UPQ 4

Definition at line 269 of file ip6_mroute.h.

Referenced by ip6_mforward().

7.41.1.10 #define MAXMIFS 64

Definition at line 76 of file ip6_mroute.h.

Referenced by add_m6if().

7.41.1.11 #define MF6C_INCOMPLETE_PARENT ((mifi_t)-1)

Definition at line 246 of file ip6_mroute.h.

Referenced by ip6_mforward().

7.41.1.12 #define MF6CHASHMOD(h) ((h) & (MF6CTBSIZ - 1))

Definition at line 264 of file ip6_mroute.h.

7.41.1.13 #define MF6CTBSIZ 256

Definition at line 262 of file ip6_mroute.h.

Referenced by expire_upcalls(), and ip6_mrouter_done().

7.41.1.14 #define MIFF_REGISTER 0x1

Definition at line 111 of file ip6_mroute.h.

Referenced by add_m6if(), del_m6if(), ip6_mdq(), and ip6_mrouter_done().

7.41.1.15 #define MRT6_ADD_MFC 104

Definition at line 59 of file ip6_mroute.h.

Referenced by ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.16 #define MRT6_ADD_MIF 102

Definition at line 57 of file ip6_mroute.h.

Referenced by ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.17 #define MRT6_DEL_MFC 105

Definition at line 60 of file ip6_mroute.h.

Referenced by ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.18 #define MRT6_DEL_MIF 103

Definition at line 58 of file ip6_mroute.h.

Referenced by ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.19 #define MRT6_DONE 101

Definition at line 56 of file ip6_mroute.h.

Referenced by ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.20 #define MRT6_INIT 108

Definition at line 62 of file ip6_mroute.h.

Referenced by ip6_mdq(), ip6_mforward(), ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.21 #define MRT6_OINIT 100

Definition at line 54 of file ip6_mroute.h.

Referenced by ip6_mdq(), ip6_mforward(), and ip6_mrouter_set().

7.41.1.22 #define MRT6_PIM 107

Definition at line 61 of file ip6_mroute.h.

Referenced by ip6_mrouter_get(), ip6_mrouter_set(), and rip6_ctloutput().

7.41.1.23 #define MRT6MSG_NOCACHE 1

Definition at line 172 of file ip6_mroute.h.

Referenced by ip6_mforward().

7.41.1.24 #define MRT6MSG_WHOLEPKT 3

Definition at line 174 of file ip6_mroute.h.

Referenced by register_send().

7.41.1.25 #define MRT6MSG_WRONGMIF 2

Definition at line 173 of file ip6_mroute.h.

Referenced by ip6_mdq().

7.41.1.26 `#define NIFBITS (sizeof(if_mask) * NBBY)`

Definition at line 83 of file ip6_mroute.h.

7.41.2 Typedef Documentation

7.41.2.1 `typedef u_int32_t if_mask`

Definition at line 82 of file ip6_mroute.h.

7.41.2.2 `typedef u_short mifi_t`

Definition at line 75 of file ip6_mroute.h.

7.41.3 Function Documentation

7.41.3.1 `int mrt6_ioctl __P ((int, caddr_t))`

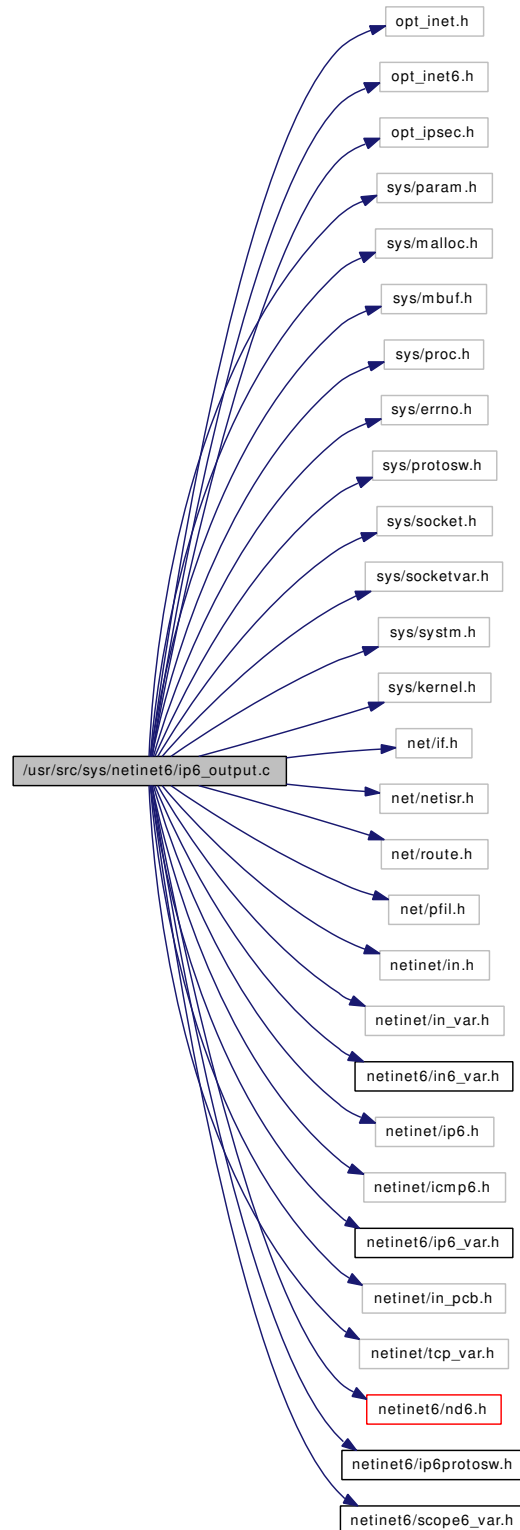
7.41.3.2 `int ip6_mrouter_done __P ((void))`

7.41.3.3 `int ip6_mrouter_get __P ((struct socket *so, struct sockopt *sopt))`

7.42 /usr/src/sys/netinet6/ip6_output.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/proc.h>
#include <sys/errno.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/system.h>
#include <sys/kernel.h>
#include <net/if.h>
#include <net/netisr.h>
#include <net/route.h>
#include <net/pfil.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet6/in6_var.h>
#include <netinet/ip6.h>
#include <netinet/icmp6.h>
#include <netinet6/ip6_var.h>
#include <netinet/in_pcb.h>
#include <netinet/tcp_var.h>
#include <netinet6/nd6.h>
#include <netinet6/ip6protosw.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for ip6_output.c:



Data Structures

- struct [ip6_exthdrs](#)

Defines

- #define [MAKE_EXTHDR](#)(hp, mp)
- #define [MAKE_CHAIN](#)(m, mp, p, i)
- #define [JUMBOOPTLEN](#) 8
- #define [OPTSET](#)(bit)
- #define [OPTSET2292](#)(bit)
- #define [OPTBIT](#)(bit) (in6p → in6p_flags & (bit) ? 1 : 0)
- #define [PKTOPT_EXTHDRCPY](#)(type)
- #define [elen](#)(x) (((struct ip6_ext *) (x)) ? (((struct ip6_ext *) (x)) → ip6e_len + 1) << 3 : 0)

Functions

- static [MALLOC_DEFINE](#) (M_IP6MOPTS, "ip6_moptions", "internet multicast options")
- static int [ip6_pcbopt](#) __P ((int, u_char *, int, struct [ip6_pktopts](#) **, int, int))
- static int [ip6_pcbopts](#) __P ((struct [ip6_pktopts](#) **, struct mbuf *, struct socket *, struct sockopt *))
- static int [ip6_getpcbopt](#) __P ((struct [ip6_pktopts](#) *, int, struct sockopt *))
- static int [ip6_setpktopt](#) __P ((int, u_char *, int, struct [ip6_pktopts](#) *, int, int, int, int))
- static int [ip6_setmoptions](#) __P ((int, struct [ip6_moptions](#) **, struct mbuf *))
- static int [ip6_getmoptions](#) __P ((int, struct [ip6_moptions](#) *, struct mbuf **))
- static int [ip6_copyexthdr](#) __P ((struct mbuf **, caddr_t, int))
- static int [ip6_insertfraghdr](#) __P ((struct mbuf *, struct mbuf *, int, struct [ip6_frag](#) **))
- static int [ip6_insert_jumboopt](#) __P ((struct [ip6_exthdrs](#) *, u_int32_t))
- static int [ip6_splthdr](#) __P ((struct mbuf *, struct [ip6_exthdrs](#) **))
- static int [ip6_getpmtu](#) __P ((struct route_in6 *, struct route_in6 *, struct ifnet *, struct [in6_addr](#) *, u_long *, int *))
- static int [copypktopts](#) __P ((struct [ip6_pktopts](#) *, struct [ip6_pktopts](#) *, int))
- int [ip6_output](#) (struct mbuf *m0, struct [ip6_pktopts](#) *opt, struct route_in6 *ro, int flags, struct [ip6_moptions](#) *im6o, struct ifnet **ifpp, struct inpcb *inpcb)
- static int [ip6_copyexthdr](#) (struct mbuf **mp, caddr_t hdr, int hlen)
- static int [ip6_insert_jumboopt](#) (struct [ip6_exthdrs](#) *exthdrs, u_int32_t plen)
- static int [ip6_insertfraghdr](#) (struct mbuf *m0, struct mbuf *m, int hlen, struct [ip6_frag](#) **frghdrp)
- static int [ip6_getpmtu](#) (struct route_in6 *ro_pmtu, struct route_in6 *ro, struct ifnet *ifp, struct [in6_addr](#) *dst, u_long *mtup, int *alwaysfragp)
- int [ip6_ctloutput](#) (struct socket *so, struct sockopt *sopt)
- int [ip6_raw_ctloutput](#) (struct socket *so, struct sockopt *sopt)
- static int [ip6_pcbopts](#) (struct [ip6_pktopts](#) **pktopt, struct mbuf *m, struct socket *so, struct sockopt *sopt)
- void [ip6_initpktopts](#) (struct [ip6_pktopts](#) *opt)
- static int [ip6_pcbopt](#) (int optname, u_char *buf, int len, struct [ip6_pktopts](#) **pktopt, int priv, int uproto)
- static int [ip6_getpcbopt](#) (struct [ip6_pktopts](#) *pktopt, int optname, struct sockopt *sopt)
- void [ip6_clearpktopts](#) (struct [ip6_pktopts](#) *pktopt, int optname)
- static int [copypktopts](#) (struct [ip6_pktopts](#) *dst, struct [ip6_pktopts](#) *src, int canwait)
- [ip6_pktopts](#) * [ip6_copypktopts](#) (struct [ip6_pktopts](#) *src, int canwait)
- void [ip6_freepcbopts](#) (struct [ip6_pktopts](#) *pktopt)
- static int [ip6_setmoptions](#) (int optname, struct [ip6_moptions](#) **im6op, struct mbuf *m)
- static int [ip6_getmoptions](#) (int optname, struct [ip6_moptions](#) *im6o, struct mbuf **mp)
- void [ip6_freemoptions](#) (struct [ip6_moptions](#) *im6o)
- int [ip6_setpktopts](#) (struct mbuf *control, struct [ip6_pktopts](#) *opt, struct [ip6_pktopts](#) *stickyopt, int priv, int uproto)

- static int `ip6_setpktopt` (int optname, u_char *buf, int len, struct `ip6_pktopts` *opt, int priv, int sticky, int cmsg, int uproto)
- void `ip6_mloopback` (struct ifnet *ifp, struct mbuf *m, struct `sockaddr_in6` *dst)
- static int `ip6_splthdr` (struct mbuf *m, struct `ip6_exthdrs` *exthdrs)
- int `ip6_optlen` (struct in6pcb *in6p)

7.42.1 Define Documentation

7.42.1.1 #define elen(x) (((struct ip6_ext *) (x)) ? (((struct ip6_ext *) (x)) -> ip6e_len + 1) << 3 : 0)

Referenced by `ip6_optlen()`, and `ip6_savecontrol()`.

7.42.1.2 #define JUMBOOPTLEN 8

Referenced by `ip6_insert_jumboopt()`.

7.42.1.3 #define MAKE_CHAIN(m, mp, p, i)

Value:

```
do {\
    if (m) {\
        if (!hdrsplit) \
            panic("assumption failed: hdr not split"); \
        *mtod(m, u_char *) = *(p); \
        *(p) = (i); \
        p = mtod(m, u_char *); \
        (m)->m_next = (mp)->m_next; \
        (mp)->m_next = (m); \
        (mp) = (m); \
    } \
} while (/*CONSTCOND*/ 0)
```

Referenced by `ip6_output()`.

7.42.1.4 #define MAKE_EXTHDR(hp, mp)

Value:

```
do {
    if (hp) {
        struct ip6_ext *eh = (struct ip6_ext *) (hp);
        error = ip6_copyexthdr((mp), (caddr_t) (hp),
            ((eh)->ip6e_len + 1) << 3);
        if (error)
            goto freehdrs;
    }
} while (/*CONSTCOND*/ 0)
```

Referenced by `ip6_output()`.

7.42.1.5 #define OPTBIT(bit) (in6p -> in6p_flags & (bit) ? 1 : 0)

Referenced by `ip6_ctloutput()`.

7.42.1.6 #define OPTSET(bit)

Value:

```
do { \
    if (optval) \
        in6p->in6p_flags |= (bit); \
    else \
        in6p->in6p_flags &= ~(bit); \
} while (/*CONSTCOND*/ 0)
```

Referenced by ip6_ctloutput().

7.42.1.7 #define OPTSET2292(bit)

Value:

```
do { \
    in6p->in6p_flags |= IN6P_RFC2292; \
    if (optval) \
        in6p->in6p_flags |= (bit); \
    else \
        in6p->in6p_flags &= ~(bit); \
} while (/*CONSTCOND*/ 0)
```

Referenced by ip6_ctloutput().

7.42.1.8 #define PKTOPT_EXTHDRCPY(type)

Value:

```
do {\
    if (src->type) {\
        int hlen = (((struct ip6_ext *)src->type)->ip6e_len + 1) << 3;\
        dst->type = malloc(hlen, M_IP6OPT, canwait);\
        if (dst->type == NULL && canwait == M_NOWAIT)\
            goto bad;\
        bcopy(src->type, dst->type, hlen);\
    }\
} while (/*CONSTCOND*/ 0)
```

Definition at line 2425 of file ip6_output.c.

Referenced by copypktopts().

7.42.2 Function Documentation

7.42.2.1 `static int copypktopts __P ((struct ip6_pktopts *, struct ip6_pktopts *, int))` [static]

7.42.2.2 `static int ip6_getpmtu __P ((struct route_in6 *, struct route_in6 *, struct ifnet *, struct in6_addr *, u_long *, int *))` [static]

7.42.2.3 `static int ip6_splithdr __P ((struct mbuf *, struct ip6_exthdrs *))` [static]

7.42.2.4 `static int ip6_insert_jumboopt __P ((struct ip6_exthdrs *, u_int32_t))` [static]

7.42.2.5 `static int ip6_insertfraghdr __P ((struct mbuf *, struct mbuf *, int, struct ip6_frag **))` [static]

7.42.2.6 `static int ip6_copyexthdr __P ((struct mbuf **, caddr_t, int))` [static]

7.42.2.7 `static int ip6_getmoptions __P ((int, struct ip6_moptions *, struct mbuf **))` [static]

7.42.2.8 `static int ip6_setmoptions __P ((int, struct ip6_moptions **, struct mbuf *))` [static]

7.42.2.9 `static int ip6_setpktopt __P ((int, u_char *, int, struct ip6_pktopts *, int, int, int, int))` [static]

7.42.2.10 `static int ip6_getpcbopt __P ((struct ip6_pktopts *, int, struct sockopt *))` [static]

7.42.2.11 `static int ip6_pcbopts __P ((struct ip6_pktopts **, struct mbuf *, struct socket *, struct sockopt *))` [static]

7.42.2.12 `static int ip6_pcbopt __P ((int, u_char *, int, struct ip6_pktopts **, int, int))` [static]

7.42.2.13 `static int copypktopts (struct ip6_pktopts * dst, struct ip6_pktopts * src, int canwait)` [static]

Definition at line 2437 of file ip6_output.c.

References ip6po_rthdr, and PKTOPT_EXTHDRCPY.

Referenced by ip6_copypktopts(), and ip6_setpktopts().

7.42.2.14 `void ip6_clearpktopts (struct ip6_pktopts * pktopt, int optname)`

Definition at line 2374 of file ip6_output.c.

References IPV6_DSTOPTS, IPV6_HOPLIMIT, IPV6_HOPOPTS, IPV6_NEXTHOP, IPV6_PKTINFO, IPV6_RTHDR, IPV6_RTHDRDSTOPTS, and IPV6_TCLASS.

Referenced by ip6_freepcbopts(), ip6_pcbopts(), ip6_setpktopt(), rip6_output(), and udp6_output().

7.42.2.15 `static int ip6_copyexthdr (struct mbuf ** mp, caddr_t hdr, int hlen)` [static]

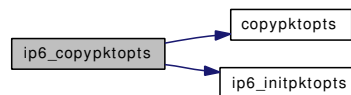
Definition at line 1175 of file ip6_output.c.

7.42.2.16 struct ip6_pktopts* ip6_copyktopts (struct ip6_pktopts * src, int canwait)

Definition at line 2482 of file ip6_output.c.

References copyktopts(), and ip6_initpktopts().

Here is the call graph for this function:



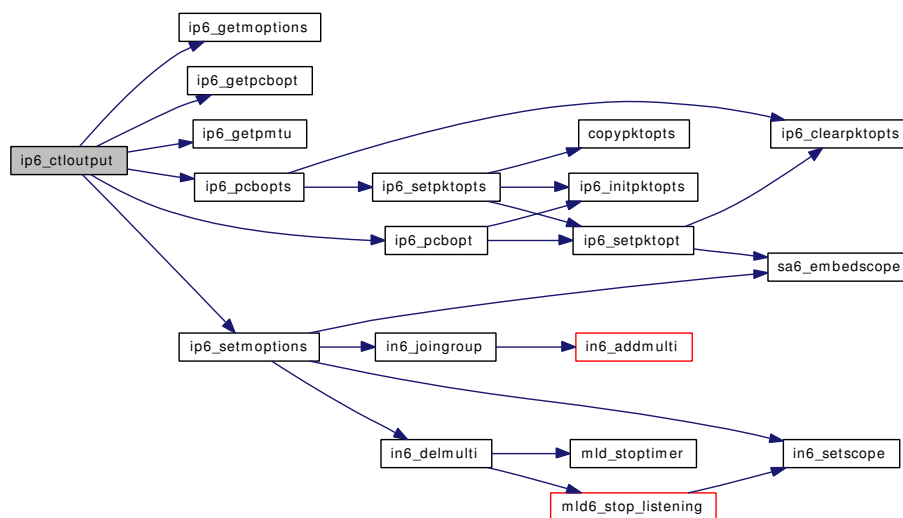
7.42.2.17 int ip6_ctloutput (struct socket * so, struct sockopt * sopt)

Definition at line 1434 of file ip6_output.c.

References IN6_IS_ADDR_UNSPECIFIED, ip6_getmoptions(), ip6_getpcbopt(), ip6_getpmtu(), ip6_pcbopt(), ip6_pcbopts(), ip6_setmoptions(), IPV6_2292DSTOPTS, IPV6_2292HOPLIMIT, IPV6_2292HOPOPTS, IPV6_2292PKTINFO, IPV6_2292PKTOPTIONS, IPV6_2292RTHDR, IPV6_AUTOFLOWLABEL, IPV6_DONTFRAG, IPV6_DSTOPTS, IPV6_FAITH, IPV6_HOPLIMIT, IPV6_HOPOPTS, IPV6_IPSEC_POLICY, IPV6_JOIN_GROUP, IPV6_LEAVE_GROUP, IPV6_MULTICAST_HOPS, IPV6_MULTICAST_IF, IPV6_MULTICAST_LOOP, IPV6_NEXTHOP, IPV6_PATHMTU, IPV6_PKTINFO, IPV6_PORTRANGE, IPV6_PORTRANGE_DEFAULT, IPV6_PORTRANGE_HIGH, IPV6_PORTRANGE_LOW, IPV6_PREFER_TEMPADDR, IPV6_RECVDSTOPTS, IPV6_RECVHOPLIMIT, IPV6_RECVHOPOPTS, IPV6_RECVPATHMTU, IPV6_RECVPKTINFO, IPV6_RECVRTHDR, IPV6_RECVRTHDRDSTOPTS, IPV6_RECVRTCLASS, IPV6_RTHDR, IPV6_RTHDRDSTOPTS, IPV6_TCLASS, IPV6_UNICAST_HOPS, IPV6_USE_MIN_MTU, IPV6_V6ONLY, OPTBIT, OPTSET, and OPTSET2292.

Referenced by ip6_ctloutput().

Here is the call graph for this function:



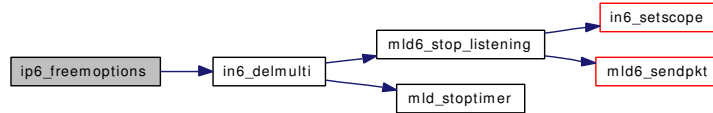
7.42.2.18 void ip6_freemoptions (struct ip6_moptions * im6o)

Definition at line 2870 of file ip6_output.c.

References in6_delmulti().

Referenced by in6_pcbfree().

Here is the call graph for this function:



7.42.2.19 void ip6_freepcbopts (struct ip6_pktopts * pktopt)

Definition at line 2503 of file ip6_output.c.

References ip6_clearpktopts().

Referenced by in6_pcbfree().

Here is the call graph for this function:



7.42.2.20 static int ip6_getmoptions (int optname, struct ip6_moptions * im6o, struct mbuf ** mp) [static]

Definition at line 2823 of file ip6_output.c.

References ip6_moptions::im6o_multicast_hlim, ip6_moptions::im6o_multicast_ifp, ip6_moptions::im6o_multicast_loop, ip6_defmcasthlim, IPV6_MULTICAST_HOPS, IPV6_MULTICAST_IF, and IPV6_MULTICAST_LOOP.

Referenced by ip6_ctloutput().

7.42.2.21 static int ip6_getpcbopt (struct ip6_pktopts * pktopt, int optname, struct sockopt * sopt) [static]

Definition at line 2273 of file ip6_output.c.

References IP6PO_DONTFRAG, IP6PO_MINMTU_MCASTONLY, IP6PO_TEMPADDR_SYSTEM, IPV6_DONTFRAG, IPV6_DSTOPTS, IPV6_HOPOPTS, IPV6_NEXTHOP, IPV6_PKTINFO, IPV6_PREFER_TEMPADDR, IPV6_RTHDR, IPV6_RTHDRDSTOPTS, IPV6_TCLASS, and IPV6_USE_MIN_MTU.

Referenced by ip6_ctloutput().

7.42.2.22 `static int ip6_getpmtu (struct route_in6 * ro_pmtu, struct route_in6 * ro, struct ifnet * ifp, struct in6_addr * dst, u_long * mtup, int * alwaysfragp)` [static]

Definition at line 1348 of file ip6_output.c.

References IN6_ARE_ADDR_EQUAL, IN6_LINKMTU, and sockaddr_in6::sin6_addr.

Referenced by ip6_ctloutput(), and ip6_output().

7.42.2.23 `void ip6_initpktopts (struct ip6_pktopts * opt)`

Definition at line 2242 of file ip6_output.c.

References IP6PO_MINMTU_MCASTONLY, and IP6PO_TEMPADDR_SYSTEM.

Referenced by ip6_copypktopts(), ip6_pcbopt(), ip6_setpktopts(), and mld6_init().

7.42.2.24 `static int ip6_insert_jumboopt (struct ip6_exthdrs * exthdrs, u_int32_t plen)` [static]

Definition at line 1208 of file ip6_output.c.

References JUMBOOPTLEN.

Referenced by ip6_output().

7.42.2.25 `static int ip6_insertfraghdr (struct mbuf * m0, struct mbuf * m, int hlen, struct ip6_frag ** frghdrp)` [static]

Definition at line 1305 of file ip6_output.c.

Referenced by ip6_output().

7.42.2.26 `void ip6_mloopback (struct ifnet * ifp, struct mbuf * m, struct sockaddr_in6 * dst)`

Definition at line 3343 of file ip6_output.c.

References in6_clearscope(), and sockaddr_in6::sin6_family.

Referenced by ip6_output(), and phyint_send().

Here is the call graph for this function:



7.42.2.27 `int ip6_optlen (struct in6pcb * in6p)`

Definition at line 3420 of file ip6_output.c.

References elen.

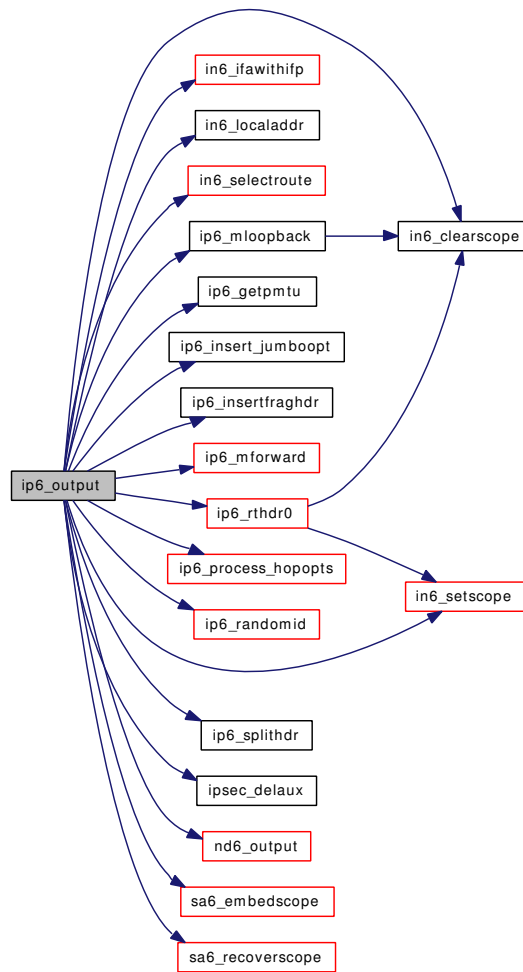
7.42.2.28 `int ip6_output (struct mbuf * m0, struct ip6_pktopts * opt, struct route_in6 * ro, int flags, struct ip6_moptions * im6o, struct ifnet ** ifpp, struct inpcb * inp)`

Definition at line 153 of file `ip6_output.c`.

References `in6_ifaddr::ia_ifa`, `ip6_moptions::im6o_multicast_hlim`, `ip6_moptions::im6o_multicast_loop`, `IN6_ARE_ADDR_EQUAL`, `in6_clearscope()`, `in6_ifawithifp()`, `in6_ifstat_inc`, `IN6_IS_ADDR_MC_INTERFACELOCAL`, `IN6_IS_ADDR_MULTICAST`, `IN6_IS_ADDR_UNSPECIFIED`, `IN6_LINKMTU`, `in6_localaddr()`, `IN6_LOOKUP_MULTI`, `in6_selectroute()`, `in6_setscope()`, `inet6_pfil_hook`, `ip6_defmcasthlim`, `ip6_getpmtu()`, `ip6_insert_jumboopt()`, `ip6_insertfraghdr()`, `ip6_mforward()`, `ip6_mloopback()`, `ip6_mrouter`, `ip6_process_hopopts()`, `ip6_randomid()`, `ip6_rthdr0()`, `ip6_splithdr()`, `ip6_use_defzone`, `ip6_pktopts::ip6po_dest1`, `ip6_pktopts::ip6po_dest2`, `IP6PO_DONTFRAG`, `ip6_pktopts::ip6po_flags`, `ip6_pktopts::ip6po_hbh`, `ip6_pktopts::ip6po_hlim`, `ip6_pktopts::ip6po_minmtu`, `IP6PO_MINMTU_ALL`, `IP6PO_MINMTU_DISABLE`, `ip6_pktopts::ip6po_tclass`, `ipsec6stat`, `ipsec_delaux()`, `IPSEC_DIR_OUTBOUND`, `IPSEC_POLICY_BYPASS`, `IPSEC_POLICY_DISCARD`, `IPSEC_POLICY_ENTRUST`, `IPSEC_POLICY_IPSEC`, `IPSEC_POLICY_NONE`, `IPV6_FORWARDING`, `IPV6_MINMTU`, `IPV6_RTHDR_TYPE_0`, `IPV6_UNSPESRC`, `M_LOOP`, `MAKE_CHAIN`, `MAKE_EXTHDR`, `nd6_output()`, `ipsecstat::out_inval`, `ipsecstat::out_polvio`, `secpolicy::policy`, `secpolicy::req`, `sa6_embedscope()`, and `sa6_recoverscope()`.

Referenced by `icmp6_redirect_output()`, `icmp6_reflect()`, `in6_gif_output()`, `ip6_forward()`, `mld6_sendpkt()`, `nd6_na_output()`, `nd6_ns_output()`, `phyint_send()`, `rip6_output()`, and `udp6_output()`.

Here is the call graph for this function:



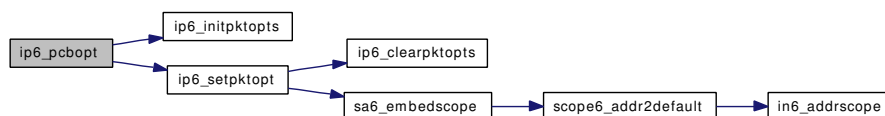
7.42.2.29 `static int ip6_pcbopt (int optname, u_char * buf, int len, struct ip6_pktopts ** pktopt, int priv, int uproto) [static]`

Definition at line 2254 of file ip6_output.c.

References ip6_initpktopts(), and ip6_setpktopt().

Referenced by ip6_ctloutput().

Here is the call graph for this function:



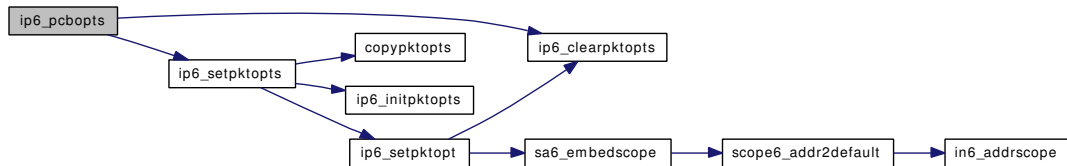
7.42.2.30 `static int ip6_pcbopts (struct ip6_pktopts ** pktopt, struct mbuf * m, struct socket * so, struct sockopt * sopt)` [static]

Definition at line 2191 of file ip6_output.c.

References ip6_clearpktopts(), ip6_setpktopts(), ip6_pktopts::ip6po_dest1, ip6_pktopts::ip6po_dest2, ip6_pktopts::ip6po_hbh, ip6_pktopts::ip6po_pktinfo, ip6po_rhinfo::ip6po_rhi_rthdr, and ip6_pktopts::ip6po_rhinfo.

Referenced by ip6_ctloutput().

Here is the call graph for this function:



7.42.2.31 `int ip6_raw_ctloutput (struct socket * so, struct sockopt * sopt)`

Definition at line 2111 of file ip6_output.c.

References in6pcb, and IPV6_CHECKSUM.

Referenced by rip6_ctloutput().

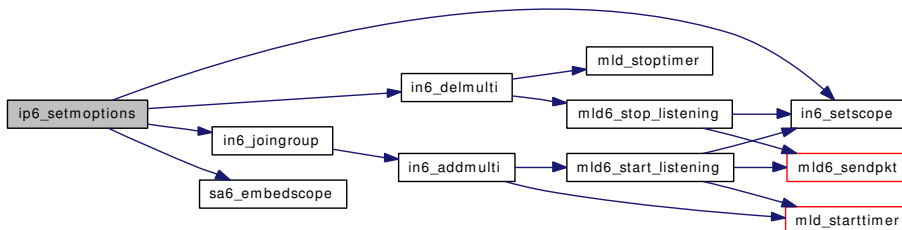
7.42.2.32 `static int ip6_setmoptions (int optname, struct ip6_moptions ** im6op, struct mbuf * m)` [static]

Definition at line 2518 of file ip6_output.c.

References in6_multi_mship::i6mm_maddr, ip6_moptions::im6o_multicast_hlim, ip6_moptions::im6o_multicast_ifp, ip6_moptions::im6o_multicast_loop, IN6_ARE_ADDR_EQUAL, in6_delmulti(), IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, in6_joining(), in6_setscope(), in6_multi::in6m_addr, in6_multi::in6m_ifp, ip6_defmcasthlim, ip6_use_defzone, IPV6_DEFAULT_MULTICAST_LOOP, IPV6_JOIN_GROUP, IPV6_LEAVE_GROUP, IPV6_MULTICAST_HOPS, IPV6_MULTICAST_IF, IPV6_MULTICAST_LOOP, ipv6_mreq::ipv6mr_interface, ipv6_mreq::ipv6mr_multiaddr, sa6_embedscope(), and sockaddr_in6::sin6_family.

Referenced by ip6_ctloutput().

Here is the call graph for this function:



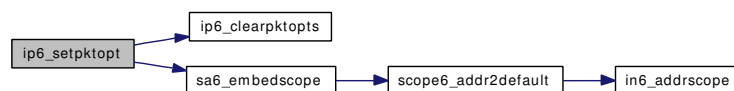
7.42.2.33 `static int ip6_setpktopt (int optname, u_char * buf, int len, struct ip6_pktopts * opt, int priv, int sticky, int cmsg, int uproto)` [static]

Definition at line 2957 of file ip6_output.c.

References IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, ip6_clearpktopts(), ip6_use_defzone, ip6_pktopts::ip6po_dest1, ip6_pktopts::ip6po_dest2, IP6PO_DONTFRAG, ip6_pktopts::ip6po_flags, ip6_pktopts::ip6po_hbh, ip6_pktopts::ip6po_hlim, ip6_pktopts::ip6po_minmtu, IP6PO_MINMTU_ALL, IP6PO_MINMTU_DISABLE, IP6PO_MINMTU_MCASTONLY, ip6_pktopts::ip6po_pktinfo, ip6_pktopts::ip6po_prefer_tempaddr, ip6_pktopts::ip6po_tclass, IP6PO_TEMPADDR_NOTPREFER, IP6PO_TEMPADDR_PREFER, IP6PO_TEMPADDR_SYSTEM, in6_pktinfo::ipi6_addr, in6_pktinfo::ipi6_ifindex, IPV6_2292DSTOPTS, IPV6_2292HOPLIMIT, IPV6_2292HOPOPTS, IPV6_2292NEXTHOP, IPV6_2292PKTINFO, IPV6_2292PKTOPTIONS, IPV6_2292RTHDR, IPV6_DONTFRAG, IPV6_DSTOPTS, IPV6_HOPLIMIT, IPV6_HOPOPTS, IPV6_NEXTHOP, IPV6_PKTINFO, IPV6_PREFER_TEMPADDR, IPV6_RTHDR, IPV6_RTHDR_TYPE_0, IPV6_RTHDRDSTOPTS, IPV6_TCLASS, IPV6_USE_MIN_MTU, sa6_embedscope(), sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_len.

Referenced by ip6_pcbopt(), and ip6_setpktopts().

Here is the call graph for this function:



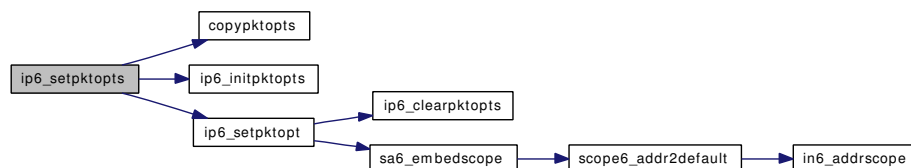
7.42.2.34 `int ip6_setpktopts (struct mbuf * control, struct ip6_pktopts * opt, struct ip6_pktopts * stickyopt, int priv, int uproto)`

Definition at line 2891 of file ip6_output.c.

References copypktopts(), ip6_initpktopts(), and ip6_setpktopt().

Referenced by ip6_pcbopts(), rip6_output(), and udp6_output().

Here is the call graph for this function:



7.42.2.35 `static int ip6_splthdr (struct mbuf * m, struct ip6_exthdrs * exthdrs)` [static]

Definition at line 3389 of file ip6_output.c.

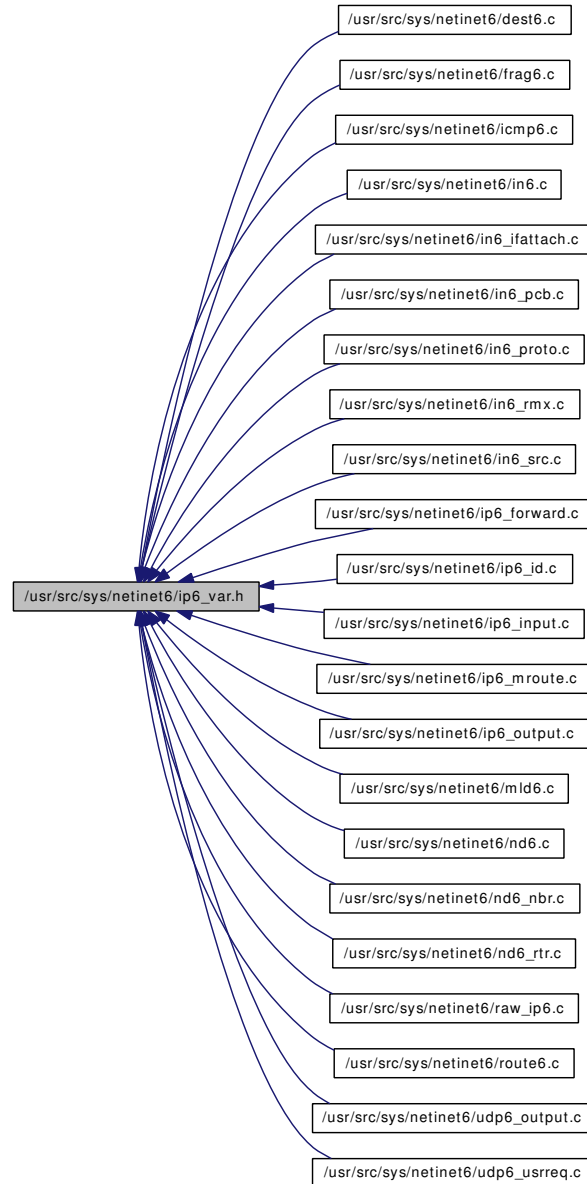
References ip6_exthdrs::ip6e_ip6.

Referenced by ip6_output().

7.42.2.36 `static MALLOC_DEFINE (M_IP6MOPTS, "ip6_moptions", "internet multicast options")` [`static`]

7.43 /usr/src/sys/netinet6/ip6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ip6q](#)
- struct [ip6asfrag](#)
- struct [ip6_moptions](#)
- struct [ip6po_rhinfo](#)
- struct [ip6po_nhinfo](#)
- struct [ip6_pktopts](#)
- struct [ip6stat](#)

- struct `ip6aux`

Defines

- #define `IP6_REASS_MBUF(ip6af)` `((struct mbuf **) &((ip6af) → ip6af_m))`
- #define `ip6po_rthdr` `ip6po_rhinfo.ip6po_rhi_rthdr`
- #define `ip6po_route` `ip6po_rhinfo.ip6po_rhi_route`
- #define `ip6po_nexthop` `ip6po_nhinfo.ip6po_nhi_nexthop`
- #define `ip6po_nextroute` `ip6po_nhinfo.ip6po_nhi_route`
- #define `IP6PO_MINMTU_MCASTONLY` -1
- #define `IP6PO_MINMTU_DISABLE` 0
- #define `IP6PO_MINMTU_ALL` 1
- #define `IP6PO_TEMPADDR_SYSTEM` -1
- #define `IP6PO_TEMPADDR_NOTPREFER` 0
- #define `IP6PO_TEMPADDR_PREFER` 1
- #define `IP6PO_DONTFRAG` 0x04
- #define `IP6PO_USECOA` 0x08
- #define `IP6A_SWAP` 0x01
- #define `IP6A_HASEEN` 0x02
- #define `IP6A_BRUID` 0x04
- #define `IP6A_RTALERTSEEN` 0x08
- #define `IPV6_UNSPECSRC` 0x01
- #define `IPV6_FORWARDING` 0x02
- #define `IPV6_MINMTU` 0x04
- #define `IP6_HDR_ALIGNED_P(ip)` `(((((intptr_t) (ip)) & 3) == 0))`

Functions

- int `icmp6_ctloutput` `__P((struct socket *, struct sockopt *sopt))`
- void `ip6_init` `__P((void))`
- void `ip6_input` `__P((struct mbuf *))`
- void `ip6_freepcbopts` `__P((struct ip6_pktopts *))`
- void `ip6_freemoptions` `__P((struct ip6_moptions *))`
- int `ip6_unknown_opt` `__P((u_int8_t *, struct mbuf *, int))`
- char *`ip6_get_prevhdr` `__P((struct mbuf *, int))`
- int `ip6_nexthdr` `__P((struct mbuf *, int, int, int *))`
- int `ip6_mforward` `__P((struct ip6_hdr *, struct ifnet *, struct mbuf *))`
- int `ip6_process_hopopts` `__P((struct mbuf *, u_int8_t *, int, u_int32_t *, u_int32_t *))`
- void `ip6_savecontrol` `__P((struct inpcb *, struct mbuf *, struct mbuf **))`
- void `ip6_notify_pmtu` `__P((struct inpcb *, struct sockaddr_in6 *, u_int32_t *))`
- int `ip6_sysctl` `__P((int *, u_int, void *, size_t *, void *, size_t))`
- void `ip6_mloopback` `__P((struct ifnet *, struct mbuf *, struct sockaddr_in6 *))`
- int `ip6_output` `__P((struct mbuf *, struct ip6_pktopts *, struct route_in6 *, int, struct ip6_moptions *, struct ifnet **, struct inpcb *))`
- int `ip6_ctloutput` `__P((struct socket *, struct sockopt *))`
- int `ip6_setpktopts` `__P((struct mbuf *, struct ip6_pktopts *, struct ip6_pktopts *, int, int))`
- void `ip6_clearpktopts` `__P((struct ip6_pktopts *, int))`
- int `ip6_optlen` `__P((struct inpcb *))`
- int `route6_input` `__P((struct mbuf **, int *, int))`

- void rip6_ctlinput __P ((int, struct sockaddr *, void *))
- int rip6_output __P ((struct mbuf *,...))
- int rip6_usreq __P ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *, struct thread *))
- in6_addr *in6_selectsrc __P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct in6_addr *, struct ifnet **, int *))
- int in6_selectroute __P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct ifnet **, struct rentry **, int))

Variables

- ip6stat ip6stat
- int ip6_defhlim
- int ip6_defmcasthlim
- int ip6_forwarding
- int ip6_forward_srcrt
- int ip6_gif_hlim
- int ip6_use_deprecated
- int ip6_rr_prune
- int ip6_mcast_pmtu
- int ip6_v6only
- socket * ip6_mrouter
- int ip6_sendredirects
- int ip6_maxfragpackets
- int ip6_maxfrags
- int ip6_sourcecheck
- int ip6_sourcecheck_interval
- int ip6_accept_rtadv
- int ip6_keepfaith
- int ip6_log_interval
- time_t ip6_log_time
- int ip6_hdrnestlimit
- int ip6_dad_count
- int ip6_auto_flowlabel
- int ip6_auto_linklocal
- int ip6_anonportmin
- int ip6_anonportmax
- int ip6_lowportmin
- int ip6_lowportmax
- int ip6_use_tempaddr
- int ip6_prefer_tempaddr
- int ip6_use_defzone
- pfil_head inet6_pfil_hook
- pr_usrreqs rip6_usrreqs

7.43.1 Define Documentation

7.43.1.1 #define IP6_HDR_ALIGNED_P(ip) (((intptr_t) (ip)) & 3) == 0)

Definition at line 286 of file ip6_var.h.

7.43.1.2 #define IP6_REASS_MBUF(ip6af) (*(struct mbuf **)&(ip6af) → ip6af_m)

Definition at line 106 of file ip6_var.h.

Referenced by frag6_freef(), and frag6_input().

7.43.1.3 #define IP6A_BRUID 0x04

Definition at line 256 of file ip6_var.h.

7.43.1.4 #define IP6A_HASEEN 0x02

Definition at line 255 of file ip6_var.h.

7.43.1.5 #define IP6A_RTALERTSEEN 0x08

Definition at line 257 of file ip6_var.h.

7.43.1.6 #define IP6A_SWAP 0x01

Definition at line 254 of file ip6_var.h.

Referenced by route6_input().

7.43.1.7 #define IP6PO_DONTFRAG 0x04

Definition at line 174 of file ip6_var.h.

Referenced by ip6_getpcbopt(), ip6_output(), and ip6_setpktopt().

7.43.1.8 #define IP6PO_MINMTU_ALL 1

Definition at line 161 of file ip6_var.h.

Referenced by ip6_output(), and ip6_setpktopt().

7.43.1.9 #define IP6PO_MINMTU_DISABLE 0

Definition at line 160 of file ip6_var.h.

Referenced by ip6_output(), and ip6_setpktopt().

7.43.1.10 #define IP6PO_MINMTU_MCASTONLY -1

Definition at line 159 of file ip6_var.h.

Referenced by ip6_getpcbopt(), ip6_initpktopts(), and ip6_setpktopt().

7.43.1.11 #define ip6po_nexthop ip6po_nhinfo.ip6po_nhi_nexthop

Definition at line 132 of file ip6_var.h.

7.43.1.12 #define ip6po_nextroute ip6po_nhinfo.ip6po_nhi_route

Definition at line 133 of file ip6_var.h.

7.43.1.13 #define ip6po_route ip6po_rhinfo.ip6po_rhi_route

Definition at line 125 of file ip6_var.h.

7.43.1.14 #define ip6po_rthdr ip6po_rhinfo.ip6po_rhi_rthdr

Definition at line 124 of file ip6_var.h.

Referenced by copypktopts().

7.43.1.15 #define IP6PO_TEMPADDR_NOTPREFER 0

Definition at line 166 of file ip6_var.h.

Referenced by ip6_setpktopt().

7.43.1.16 #define IP6PO_TEMPADDR_PREFER 1

Definition at line 167 of file ip6_var.h.

Referenced by ip6_setpktopt().

7.43.1.17 #define IP6PO_TEMPADDR_SYSTEM -1

Definition at line 165 of file ip6_var.h.

Referenced by ip6_getpcbopt(), ip6_initpktopts(), and ip6_setpktopt().

7.43.1.18 #define IP6PO_USECOA 0x08

Definition at line 175 of file ip6_var.h.

7.43.1.19 #define IPV6_FORWARDING 0x02

Definition at line 280 of file ip6_var.h.

Referenced by ip6_forward(), ip6_output(), and phyint_send().

7.43.1.20 #define IPV6_MINMTU 0x04

Definition at line 281 of file ip6_var.h.

Referenced by in6_gif_output(), and ip6_output().

7.43.1.21 `#define IPV6_UNSPECSRC 0x01`

Definition at line 279 of file ip6_var.h.

Referenced by ip6_output(), and nd6_ns_output().

7.43.2 Function Documentation

- 7.43.2.1 `int in6_selectroute __P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct ifnet **, struct rtenry **, int))`
- 7.43.2.2 `struct in6_addr* in6_selectsrc __P ((struct sockaddr_in6 *, struct ip6_pktopts *, struct ip6_moptions *, struct route_in6 *, struct in6_addr *, struct ifnet **, int *))`
- 7.43.2.3 `int rip6_usrreq __P ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *, struct thread *))`
- 7.43.2.4 `int rip6_output __P ((struct mbuf *,...))`
- 7.43.2.5 `void rip6_ctlinput __P ((int, struct sockaddr *, void *))`
- 7.43.2.6 `int route6_input __P ((struct mbuf **, int *, int))`
- 7.43.2.7 `int ip6_optlen __P ((struct inpcb *))`
- 7.43.2.8 `struct ip6_pktopts *ip6_copypktopts __P ((struct ip6_pktopts *, int))`
- 7.43.2.9 `int ip6_setpktopts __P ((struct mbuf *, struct ip6_pktopts *, struct ip6_pktopts *, int, int))`
- 7.43.2.10 `int rip6_ctloutput __P ((struct socket *, struct sockopt *))`
- 7.43.2.11 `int ip6_output __P ((struct mbuf *, struct ip6_pktopts *, struct route_in6 *, int, struct ip6_moptions *, struct ifnet **, struct inpcb *))`
- 7.43.2.12 `void ip6_mloopback __P ((struct ifnet *, struct mbuf *, struct sockaddr_in6 *))`
- 7.43.2.13 `int ip6_sysctl __P ((int *, u_int, void *, size_t *, void *, size_t))`
- 7.43.2.14 `void ip6_notify_pmtu __P ((struct inpcb *, struct sockaddr_in6 *, u_int32_t *))`
- 7.43.2.15 `void ip6_savecontrol __P ((struct inpcb *, struct mbuf *, struct mbuf **))`
- 7.43.2.16 `int ip6_process_hopopts __P ((struct mbuf *, u_int8_t *, int, u_int32_t *, u_int32_t *))`
- 7.43.2.17 `int ip6_mforward __P ((struct ip6_hdr *, struct ifnet *, struct mbuf *))`
- 7.43.2.18 `int ip6_lasthdr __P ((struct mbuf *, int, int, int *))`
- 7.43.2.19 `char* ip6_get_prevhdr __P ((struct mbuf *, int))`
- 7.43.2.20 `int ip6_unknown_opt __P ((u_int8_t *, struct mbuf *, int))`
- 7.43.2.21 `void ip6_freemoptions __P ((struct ip6_moptions *))`
- 7.43.2.22 `void ip6_initpktopts __P ((struct ip6_pktopts *))`
- 7.43.2.23 `void ipsec_clearhist __P ((struct mbuf *))`
- 7.43.2.24 `void ip6_init __P ((void))`
- 7.43.2.25 `int icmp6_ctloutput __P ((struct socket *, struct sockopt *, sockopt*))`

Generated on Sat Feb 24 19:41:02 2007 for FreeBSD kernel IPv6 code by Doxygen

7.43.3 Variable Documentation

- 7.43.3.1 `struct pfil_head inet6_pfil_hook`

Referenced by `ip6_forward()`, `ip6_input()`, and `ip6_output()`.

7.43.3.2 `int ip6_accept_rtadv`

Definition at line 403 of file `in6_proto.c`.

Referenced by `defrouter_select()`, `defrtrlist_del()`, `nd6_cache_lladdr()`, `nd6_ra_input()`, and `nd6_rs_input()`.

7.43.3.3 `int ip6_anonportmax`

7.43.3.4 `int ip6_anonportmin`

7.43.3.5 `int ip6_auto_flowlabel`

Definition at line 409 of file `in6_proto.c`.

7.43.3.6 `int ip6_auto_linklocal`

Definition at line 66 of file `in6_ifattach.c`.

Referenced by `in6_ifattach()`.

7.43.3.7 `int ip6_dad_count`

Definition at line 408 of file `in6_proto.c`.

Referenced by `nd6_dad_start()`.

7.43.3.8 `int ip6_defhlim`

Definition at line 401 of file `in6_proto.c`.

Referenced by `icmp6_reflect()`, and `in6_selecthlim()`.

7.43.3.9 `int ip6_defmcasthlim`

Definition at line 402 of file `in6_proto.c`.

Referenced by `ip6_getmoptions()`, `ip6_output()`, and `ip6_setmoptions()`.

7.43.3.10 `int ip6_forward_srcrt`

Definition at line 128 of file `ip6_input.c`.

7.43.3.11 `int ip6_forwarding`

Definition at line 399 of file `in6_proto.c`.

Referenced by defrouter_select(), defrtrlist_del(), icmp6_redirect_input(), icmp6_redirect_output(), ip6_input(), nd6_cache_lladdr(), nd6_free(), nd6_is_new_addr_neighbor(), nd6_na_input(), nd6_ns_input(), and nd6_rs_input().

7.43.3.12 int ip6_gif_hlim

Definition at line 410 of file in6_proto.c.

Referenced by in6_gif_output().

7.43.3.13 int ip6_hdrnestlimit

Definition at line 407 of file in6_proto.c.

Referenced by ip6_input().

7.43.3.14 int ip6_keepfaith

Definition at line 417 of file in6_proto.c.

Referenced by ip6_input().

7.43.3.15 int ip6_log_interval

Definition at line 406 of file in6_proto.c.

Referenced by ip6_forward(), and ip6_mforward().

7.43.3.16 time_t ip6_log_time

Definition at line 418 of file in6_proto.c.

Referenced by ip6_forward(), and ip6_mforward().

7.43.3.17 int ip6_lowportmax

7.43.3.18 int ip6_lowportmin

7.43.3.19 int ip6_maxfragpackets

Definition at line 404 of file in6_proto.c.

Referenced by frag6_change(), frag6_init(), frag6_input(), and frag6_slowtimo().

7.43.3.20 int ip6_maxfrags

Definition at line 405 of file in6_proto.c.

Referenced by frag6_change(), frag6_init(), and frag6_input().

7.43.3.21 int ip6_mcast_pmtu

Definition at line 414 of file in6_proto.c.

Referenced by phyint_send().

7.43.3.22 struct socket* ip6_mrouter

Definition at line 137 of file ip6_mroute.c.

Referenced by ip6_input(), ip6_output(), mld6_sendpkt(), and rip6_detach().

7.43.3.23 int ip6_prefer_tempaddr

Definition at line 113 of file in6_src.c.

7.43.3.24 int ip6_rr_prune

Definition at line 412 of file in6_proto.c.

7.43.3.25 int ip6_sendredirects

Definition at line 400 of file in6_proto.c.

Referenced by ip6_forward().

7.43.3.26 int ip6_sourcecheck

Definition at line 129 of file ip6_input.c.

7.43.3.27 int ip6_sourcecheck_interval

Definition at line 130 of file ip6_input.c.

7.43.3.28 int ip6_use_defzone

Definition at line 52 of file scope6.c.

Referenced by in6_pcbbind(), in6_pcblladdr(), ip6_output(), ip6_setmoptions(), ip6_setpktopt(), rip6_bind(), rip6_connect(), rip6_output(), udp6_getcred(), and udp6_output().

7.43.3.29 int ip6_use_deprecated

Definition at line 411 of file in6_proto.c.

Referenced by in6_ifawithifp(), and in6_selectsrc().

7.43.3.30 int `ip6_use_tempaddr`

Definition at line 90 of file `nd6_rtr.c`.

Referenced by `in6_control()`, and `nd6_timer()`.

7.43.3.31 int `ip6_v6only`

Definition at line 415 of file `in6_proto.c`.

Referenced by `sctp6_connect()`, and `sctp6_send()`.

7.43.3.32 struct `ip6stat ip6stat`

Definition at line 136 of file `ip6_input.c`.

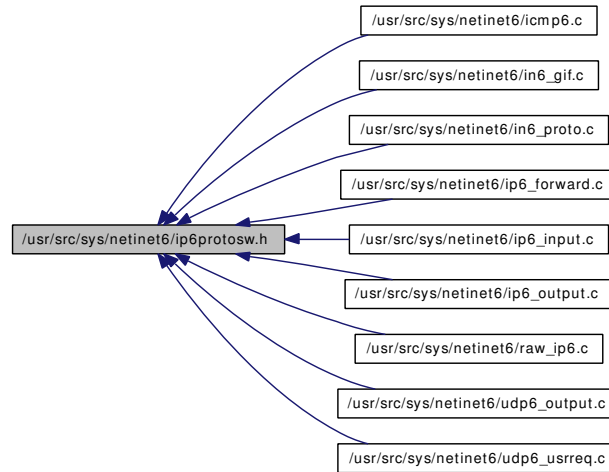
Referenced by `ip6_hopopts_input()`, `ip6_input()`, `ip6_process_hopopts()`, `ip6_savecontrol()`, and `ip6_unknown_opt()`.

7.43.3.33 struct `pr_usrreqs rip6_usrreqs`

Definition at line 802 of file `raw_ip6.c`.

7.44 /usr/src/sys/netinet6/ip6protosw.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ip6ctlparam](#)
- struct [ip6protosw](#)

Variables

- [ip6protosw inet6sw](#) []

7.44.1 Variable Documentation

7.44.1.1 struct [ip6protosw inet6sw](#) []

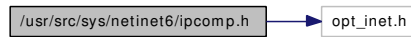
Definition at line 160 of file `in6_proto.c`.

Referenced by `icmp6_notify_error()`, `ip6_init()`, and `ip6_input()`.

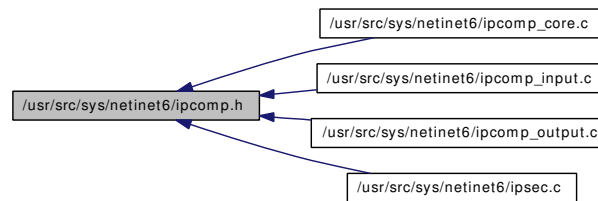
7.45 /usr/src/sys/netinet6/ipcomp.h File Reference

```
#include "opt_inet.h"
```

Include dependency graph for ipcomp.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ipcomp](#)
- struct [ipcomp_algorithm](#)

Defines

- #define [IPCOMP_OUI](#) 1
- #define [IPCOMP_DEFLATE](#) 2
- #define [IPCOMP_LZS](#) 3
- #define [IPCOMP_MAX](#) 4
- #define [IPCOMP_CPI_NEGOTIATE_MIN](#) 256

Functions

- [ipcomp_algorithm](#) *ipcomp_algorithm_lookup __P ((int))
- void ipcomp4_input __P ((struct mbuf *, int))
- int ipcomp4_output __P ((struct mbuf *, struct [ipsecrequest](#) *))

7.45.1 Define Documentation

7.45.1.1 #define IPCOMP_CPI_NEGOTIATE_MIN 256

Definition at line 56 of file ipcomp.h.

7.45.1.2 #define IPCOMP_DEFLATE 2

Definition at line 52 of file ipcomp.h.

7.45.1.3 `#define IPCOMP_LZS 3`

Definition at line 53 of file ipcomp.h.

7.45.1.4 `#define IPCOMP_MAX 4`

Definition at line 54 of file ipcomp.h.

7.45.1.5 `#define IPCOMP_OUI 1`

Definition at line 51 of file ipcomp.h.

7.45.2 Function Documentation

7.45.2.1 `int ipcomp4_output __P ((struct mbuf *, struct ipsecrequest *))`

7.45.2.2 `void ipcomp4_input __P ((struct mbuf *, int))`

7.45.2.3 `struct ipcomp_algorithm* ipcomp_algorithm_lookup __P ((int))`

7.46 /usr/src/sys/netinet6/ipcomp6.h File Reference

Functions

- int ipcomp6_input __P ((struct mbuf **, int *, int))
- int ipcomp6_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))

7.46.1 Function Documentation

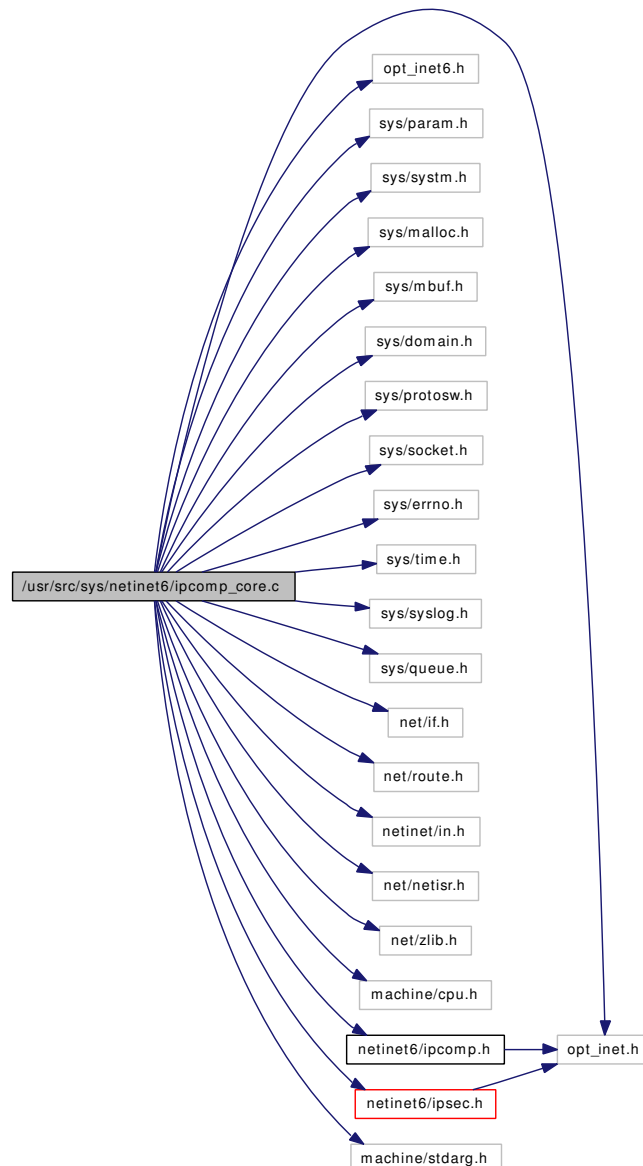
7.46.1.1 int ipcomp6_output __P ((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *))

7.46.1.2 int ipcomp6_input __P ((struct mbuf **, int *, int))

7.47 /usr/src/sys/netinet6/ipcomp_core.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/malloc.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <sys/queue.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <net/netisr.h>  
#include <net/zlib.h>  
#include <machine/cpu.h>  
#include <netinet6/ipcomp.h>  
#include <netinet6/ipsec.h>  
#include <machine/stdarg.h>
```

Include dependency graph for ipcomp_core.c:



Defines

- #define [MOREBLOCK\(\)](#)

Functions

- static void *deflate_alloc __P ((void *, u_int, u_int))
- static void deflate_free __P ((void *, void *))
- static int deflate_common __P ((struct mbuf *, struct mbuf *, size_t *, int))
- static int deflate_compress __P ((struct mbuf *, struct mbuf *, size_t *))
- ipcomp_algorithm * ipcomp_algorithm_lookup (int idx)
- static void * deflate_alloc (void *aux, u_int items, u_int siz)
- static void deflate_free (void *aux, void *ptr)

- static int `deflate_common` (struct mbuf *m, struct mbuf *md, [size_t](#) *lenp, int mode)
- static int `deflate_compress` (struct mbuf *m, struct mbuf *md, [size_t](#) *lenp)
- static int `deflate_decompress` (struct mbuf *m, struct mbuf *md, [size_t](#) *lenp)

Variables

- static int `deflate_policy` = Z_DEFAULT_COMPRESSION
- static int `deflate_window_out` = -12
- static const int `deflate_window_in` = -1 * MAX_WBITS
- static int `deflate_memlevel` = MAX_MEM_LEVEL
- static struct `ipcomp_algorithm` `ipcomp_algorithms` []

7.47.1 Define Documentation

7.47.1.1 #define MOREBLOCK()

Referenced by `deflate_common()`.

7.47.2 Function Documentation

7.47.2.1 static int `deflate_decompress` __P ((struct mbuf *, struct mbuf *, [size_t](#) *)) [static]

7.47.2.2 static int `deflate_common` __P ((struct mbuf *, struct mbuf *, [size_t](#) *, int)) [static]

7.47.2.3 static void `deflate_free` __P ((void *, void *)) [static]

7.47.2.4 static void* `deflate_alloc` __P ((void *, u_int, u_int)) [static]

7.47.2.5 static void* `deflate_alloc` (void *aux, u_int items, u_int siz) [static]

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

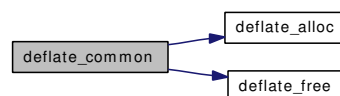
Referenced by `deflate_common()`.

7.47.2.6 static int `deflate_common` (struct mbuf * m, struct mbuf * md, [size_t](#) * lenp, int mode) [static]

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

References `deflate_alloc()`, `deflate_free()`, `ipseclog`, and `MOREBLOCK`.

Here is the call graph for this function:



7.47.2.7 `static int deflate_compress (struct mbuf * m, struct mbuf * md, size_t * lenp)` [static]

Definition at line 324 of file ipcomp_core.c.

7.47.2.8 `static int deflate_decompress (struct mbuf * m, struct mbuf * md, size_t * lenp)`
[static]

Definition at line 340 of file ipcomp_core.c.

7.47.2.9 `static void deflate_free (void * aux, void * ptr)` [static]

Definition at line 113 of file ipcomp_core.c.

Referenced by deflate_common().

7.47.2.10 `struct ipcomp_algorithm* ipcomp_algorithm_lookup (int idx)`

Definition at line 92 of file ipcomp_core.c.

References ipcomp_algorithms.

Referenced by ipcomp_output().

7.47.3 Variable Documentation

7.47.3.1 `int deflate_memlevel = MAX_MEM_LEVEL` [static]

Definition at line 85 of file ipcomp_core.c.

7.47.3.2 `int deflate_policy = Z_DEFAULT_COMPRESSION` [static]

Definition at line 82 of file ipcomp_core.c.

7.47.3.3 `const int deflate_window_in = -1 * MAX_WBITS` [static]

Definition at line 84 of file ipcomp_core.c.

7.47.3.4 `int deflate_window_out = -12` [static]

Definition at line 83 of file ipcomp_core.c.

7.47.3.5 `struct ipcomp_algorithm ipcomp_algorithms[]` [static]

Initial value:

```
{
    { deflate_compress, deflate_decompress, 90 },
}
```

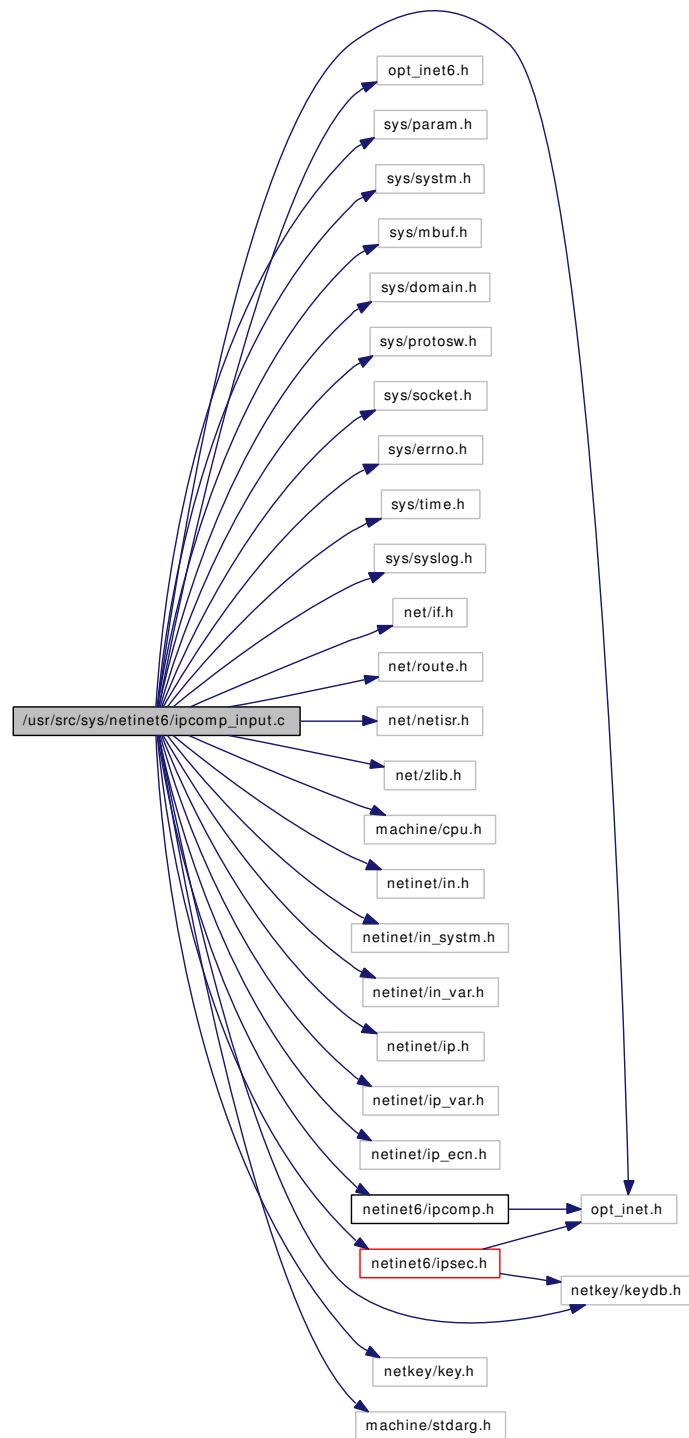
Definition at line 87 of file ipcomp_core.c.

Referenced by ipcomp_algorithm_lookup().

7.48 /usr/src/sys/netinet6/ipcomp_input.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <net/netisr.h>  
#include <net/zlib.h>  
#include <machine/cpu.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/in_var.h>  
#include <netinet/ip.h>  
#include <netinet/ip_var.h>  
#include <netinet/ip_ecn.h>  
#include <netinet6/ipcomp.h>  
#include <netinet6/ipsec.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>  
#include <machine/stdarg.h>
```

Include dependency graph for ipcomp_input.c:



Defines

- `#define` [IPLEN_FLIPPED](#)

7.48.1 Define Documentation

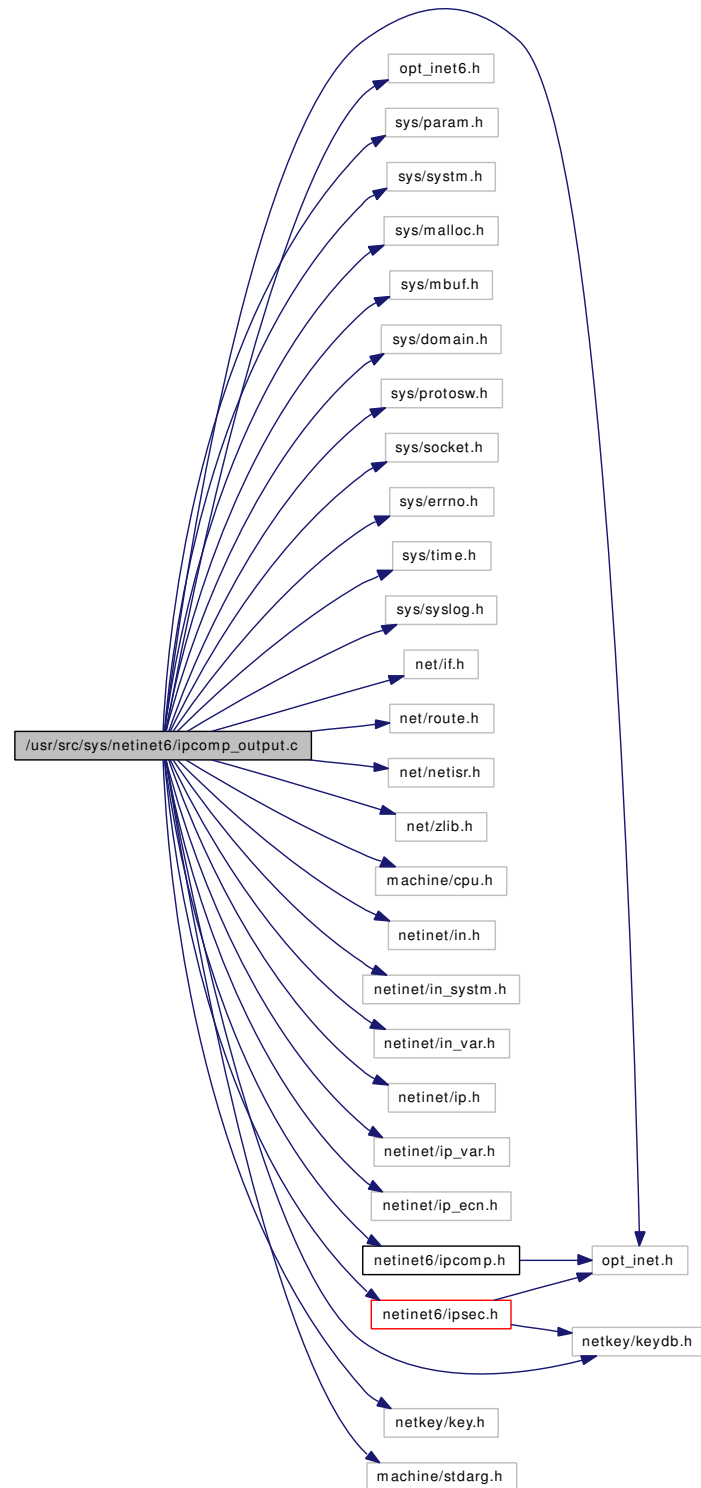
7.48.1.1 #define IPLEN_FLIPPED

Definition at line 81 of file ipcomp_input.c.

7.49 /usr/src/sys/netinet6/ipcomp_output.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/malloc.h>  
#include <sys/mbuf.h>  
#include <sys/domain.h>  
#include <sys/protosw.h>  
#include <sys/socket.h>  
#include <sys/errno.h>  
#include <sys/time.h>  
#include <sys/syslog.h>  
#include <net/if.h>  
#include <net/route.h>  
#include <net/netisr.h>  
#include <net/zlib.h>  
#include <machine/cpu.h>  
#include <netinet/in.h>  
#include <netinet/in_system.h>  
#include <netinet/in_var.h>  
#include <netinet/ip.h>  
#include <netinet/ip_var.h>  
#include <netinet/ip_ecn.h>  
#include <netinet6/ipcomp.h>  
#include <netinet6/ipsec.h>  
#include <netkey/key.h>  
#include <netkey/keydb.h>  
#include <machine/stdarg.h>
```

Include dependency graph for ipcomp_output.c:



Functions

- static int ipcomp_output __P((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *, int))
- static int ipcomp_output (struct mbuf *m, u_char *nextthrp, struct mbuf *md, struct ipsecrequest

*isr, int af)

7.49.1 Function Documentation

7.49.1.1 `static int ipcomp_output __P((struct mbuf *, u_char *, struct mbuf *, struct ipsecrequest *, int))` [static]

7.49.1.2 `static int ipcomp_output (struct mbuf * m, u_char * nexthdrp, struct mbuf * md, struct ipsecrequest * isr, int af)` [static]

Definition at line 104 of file ipcomp_output.c.

References ipcomp_algorithm_lookup(), ipsec6stat, ipseclog, ipsecstat::out_inval, and ipsecrequest::sav.

Here is the call graph for this function:

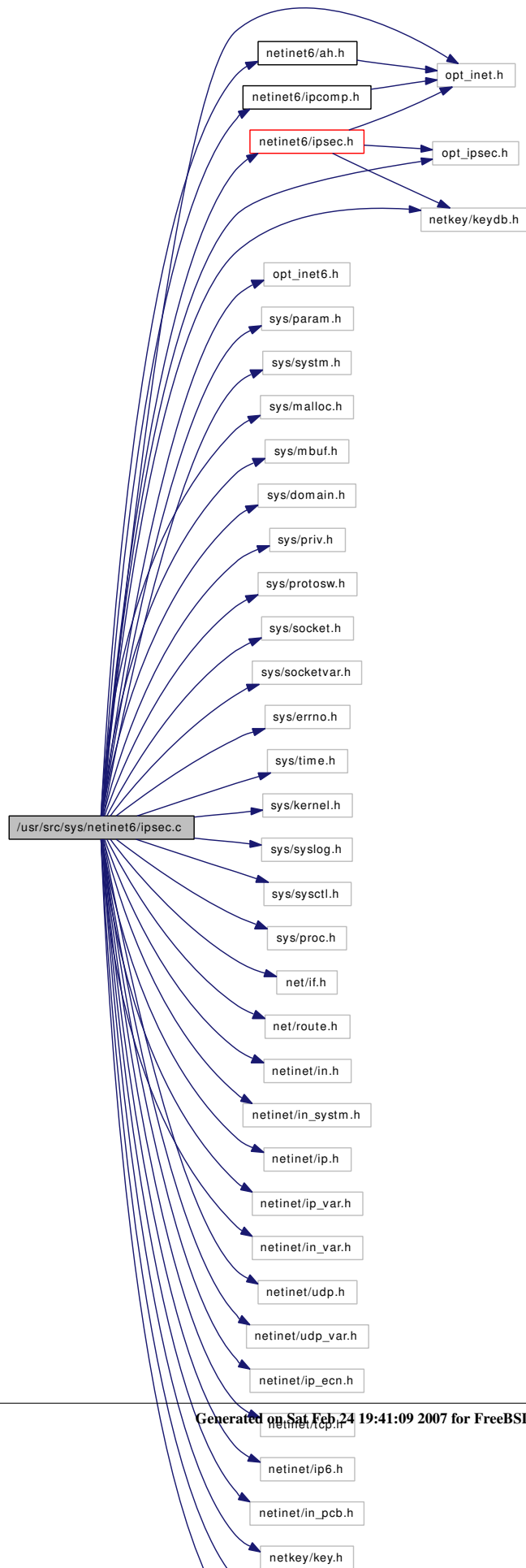


7.50 /usr/src/sys/netinet6/ipsec.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <sys/sysctl.h>
#include <sys/proc.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/in_var.h>
#include <netinet/udp.h>
#include <netinet/udp_var.h>
#include <netinet/ip_ecn.h>
#include <netinet/tcp.h>
#include <netinet/ip6.h>
#include <netinet/in_pcb.h>
#include <netinet6/ipsec.h>
#include <netinet6/ah.h>
#include <netinet6/ipcomp.h>
#include <netkey/key.h>
```

```
#include <netkey/keydb.h>
#include <netkey/key_debug.h>
#include <machine/in_cksum.h>
```

Include dependency graph for ipsec.c:



Functions

- `NET_NEEDS_GIANT` ("ipsec")
- `SYSCTL_DECL` (`_net_inet_ipsec`)
- `SYSCTL_STRUCT` (`_net_inet_ipsec`, `IPSECCTL_STATS`, `stats`, `CTLFLAG_RD`, `&ipsecstat`, `ipsecstat`, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DEF_ESP_TRANSLEV`, `esp_trans_deflev`, `CTLFLAG_RW`, `&ip4_esp_trans_deflev`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DEF_ESP_NETLEV`, `esp_net_deflev`, `CTLFLAG_RW`, `&ip4_esp_net_deflev`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DEF_AH_TRANSLEV`, `ah_trans_deflev`, `CTLFLAG_RW`, `&ip4_ah_trans_deflev`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DEF_AH_NETLEV`, `ah_net_deflev`, `CTLFLAG_RW`, `&ip4_ah_net_deflev`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_AH_CLEARRTOS`, `ah_clearartos`, `CTLFLAG_RW`, `&ip4_ah_clearartos`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_AH_OFFSETMASK`, `ah_offsetmask`, `CTLFLAG_RW`, `&ip4_ah_offsetmask`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DFBIT`, `dfbit`, `CTLFLAG_RW`, `&ip4_ipsec_dfbit`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_ECN`, `ecn`, `CTLFLAG_RW`, `&ip4_ipsec_ecn`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_DEBUG`, `debug`, `CTLFLAG_RW`, `&ipsec_debug`, 0, "")
- `SYSCTL_INT` (`_net_inet_ipsec`, `IPSECCTL_ESP_RANDPAD`, `esp_randpad`, `CTLFLAG_RW`, `&ip4_esp_randpad`, 0, "")
- `static struct secpolicy *ipsec_checkpcbcache` `__P` ((`struct mbuf *`, `struct inpcbpolicy *`, `int`))
- `static int ipsec_fillpcbcache` `__P` ((`struct inpcbpolicy *`, `struct mbuf *`, `struct secpolicy *`, `int`))
- `static int ipsec_invalpcbcache` `__P` ((`struct inpcbpolicy *`, `int`))
- `static int ipsec_setspidx_mbuf` `__P` ((`struct secpolicyindex *`, `int`, `struct mbuf *`, `int`))
- `static int ipsec_setspidx` `__P` ((`struct mbuf *`, `struct secpolicyindex *`, `int`))
- `static int ipsec4_setspidx_ipaddr` `__P` ((`struct mbuf *`, `struct secpolicyindex *`))
- `static struct inpcbpolicy *ipsec_newpcbpolicy` `__P` ((`void`))
- `static void ipsec_delpcbpolicy` `__P` ((`struct inpcbpolicy *`))
- `static struct secpolicy *ipsec_deepcopy_policy` `__P` ((`struct secpolicy *`))
- `static int ipsec_set_policy` `__P` ((`struct secpolicy **`, `int`, `caddr_t`, `size_t`, `int`))
- `static int ipsec_get_policy` `__P` ((`struct secpolicy *`, `struct mbuf **`))
- `static void vshiftl` `__P` ((`unsigned char *`, `int`, `int`))
- `static int ipsec_in_reject` `__P` ((`struct secpolicy *`, `struct mbuf *`))
- `static struct ipsecaux *ipsec_addaux` `__P` ((`struct mbuf *`))
- `static void ipsec_optaux` `__P` ((`struct mbuf *`, `struct ipsecaux *`))
- `static struct secpolicy *ipsec_checkpcbcache` (`struct mbuf *m`, `struct inpcbpolicy *pcbsp`, `int dir`)
- `static int ipsec_fillpcbcache` (`struct inpcbpolicy *pcbsp`, `struct mbuf *m`, `struct secpolicy *sp`, `int dir`)
- `static int ipsec_invalpcbcache` (`struct inpcbpolicy *pcbsp`, `int dir`)
- `int ipsec_pcbconn` (`struct inpcbpolicy *pcbsp`)
- `int ipsec_pcbdisconn` (`struct inpcbpolicy *pcbsp`)
- `int ipsec_invalpcbcacheall` ()
- `secpolicy * ipsec4_getpolicybypcb` (`struct mbuf *m`, `u_int dir`, `struct inpcb *inp`, `int *error`)
- `secpolicy * ipsec4_getpolicybyaddr` (`struct mbuf *m`, `u_int dir`, `int flag`, `int *error`)
- `int ipsec_setspidx_mbuf` (`struct secpolicyindex *spidx`, `int family`, `struct mbuf *m`, `int needport`)
- `static int ipsec_setspidx` (`struct mbuf *m`, `struct secpolicyindex *spidx`, `int needport`)
- `static void ipsec4_get_ulp` (`struct mbuf *m`, `struct secpolicyindex *spidx`, `int needport`)
- `static int ipsec4_setspidx_ipaddr` (`struct mbuf *m`, `struct secpolicyindex *spidx`)
- `static struct inpcbpolicy * ipsec_newpcbpolicy` ()

- static void `ipsec_delpcbpolicy` (struct `inpcbpolicy` *p)
- int `ipsec_init_pcbpolicy` (struct socket *so, struct `inpcbpolicy` **pcb_sp)
- int `ipsec_copy_pcbpolicy` (struct `inpcbpolicy` *old, struct `inpcbpolicy` *new)
- static struct `secpolicy` * `ipsec_deepcopy_policy` (struct `secpolicy` *src)
- static int `ipsec_set_policy` (struct `secpolicy` **spp, int optname, caddr_t request, `size_t` len, int priv)
- static int `ipsec_get_policy` (struct `secpolicy` *sp, struct mbuf **mp)
- int `ipsec4_set_policy` (struct inpcb *inp, int optname, caddr_t request, `size_t` len, int priv)
- int `ipsec4_get_policy` (struct inpcb *inp, caddr_t request, `size_t` len, struct mbuf **mp)
- int `ipsec4_delete_pcbpolicy` (struct inpcb *inp)
- u_int `ipsec_get_reqlevel` (struct `ipsecrequest` *isr, int af)
- static int `ipsec_in_reject` (struct `secpolicy` *sp, struct mbuf *m)
- int `ipsec4_in_reject` (struct mbuf *m, struct inpcb *inp)
- static `size_t` `ipsec_hdrsiz` (struct `secpolicy` *sp)
- `size_t` `ipsec4_hdrsiz` (struct mbuf *m, u_int dir, struct inpcb *inp)
- int `ipsec_chkreplay` (u_int32_t seq, struct `secasvar` *sav)
- int `ipsec_updatereplay` (u_int32_t seq, struct `secasvar` *sav)
- static void `vshiffl` (unsigned char *bitmap, int nbit, int wsize)
- const char * `ipsec4_logpacketstr` (struct ip *ip, u_int32_t spi)
- const char * `ipsec_logsastr` (struct `secasvar` *sav)
- void `ipsec_dumpmbuf` (struct mbuf *m)
- int `ipsec4_tunnel_validate` (struct mbuf *m, int off, u_int nxt0, struct `secasvar` *sav)
- mbuf * `ipsec_copypkt` (struct mbuf *m)
- static struct `ipsecaux` * `ipsec_addaux` (struct mbuf *m)
- static struct `ipsecaux` * `ipsec_findaux` (struct mbuf *m)
- void `ipsec_delaux` (struct mbuf *m)
- static void `ipsec_optaux` (struct mbuf *m, struct `ipsecaux` *aux)
- int `ipsec_addhist` (struct mbuf *m, int proto, u_int32_t spi)
- int `ipsec_getnhist` (struct mbuf *m)
- void `ipsec_clearhist` (struct mbuf *m)

Variables

- int `ipsec_debug` = 0
- `ipsecstat` `ipsecstat`
- int `ip4_ah_clearartos` = 1
- int `ip4_ah_offsetmask` = 0
- int `ip4_ipsec_dfbit` = 0
- int `ip4_esp_trans_deflev` = IPSEC_LEVEL_USE
- int `ip4_esp_net_deflev` = IPSEC_LEVEL_USE
- int `ip4_ah_trans_deflev` = IPSEC_LEVEL_USE
- int `ip4_ah_net_deflev` = IPSEC_LEVEL_USE
- `secpolicy` * `ip4_def_policy`
- int `ip4_ipsec_ecn` = 0
- int `ip4_esp_randpad` = -1
- static int `sp_cachegen` = 1

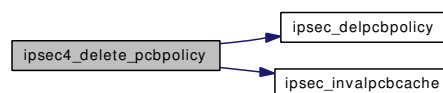
7.50.1 Function Documentation

- 7.50.1.1** `static void ipsec_optaux __P ((struct mbuf *, struct ipsecaux *))` [static]
- 7.50.1.2** `static struct ipsecaux* ipsec_addaux __P ((struct mbuf *))` [static]
- 7.50.1.3** `static int ipsec_in_reject __P ((struct secpolicy *, struct mbuf *))` [static]
- 7.50.1.4** `static void vshiffl __P ((unsigned char *, int, int))` [static]
- 7.50.1.5** `static int ipsec_get_policy __P ((struct secpolicy *, struct mbuf **))` [static]
- 7.50.1.6** `static int ipsec_set_policy __P ((struct secpolicy **, int, caddr_t, size_t, int))` [static]
- 7.50.1.7** `static size_t ipsec_hdrsiz __P ((struct secpolicy *))` [static]
- 7.50.1.8** `static void ipsec_delpcbpolicy __P ((struct inpcbpolicy *))` [static]
- 7.50.1.9** `static struct inpcbpolicy* ipsec_newpcbpolicy __P ((void))` [static]
- 7.50.1.10** `static int ipsec4_setspidx_ipaddr __P ((struct mbuf *, struct secpolicyindex *))`
[static]
- 7.50.1.11** `static void ipsec4_get_ulp __P ((struct mbuf *, struct secpolicyindex *, int))` [static]
- 7.50.1.12** `static int ipsec_setspidx_mbuf __P ((struct secpolicyindex *, int, struct mbuf *, int))`
[static]
- 7.50.1.13** `static int ipsec_invalpcbcache __P ((struct inpcbpolicy *, int))` [static]
- 7.50.1.14** `static int ipsec_fillpcbcache __P ((struct inpcbpolicy *, struct mbuf *, struct secpolicy *, int))` [static]
- 7.50.1.15** `static struct secpolicy* ipsec_checkpcbcache __P ((struct mbuf *, struct inpcbpolicy *, int))` [static]
- 7.50.1.16** `int ipsec4_delete_pcbpolicy (struct inpcb *inp)`

Definition at line 1517 of file ipsec.c.

References `ipsec_delpcbpolicy()`, `IPSEC_DIR_ANY`, and `ipsec_invalpcbcache()`.

Here is the call graph for this function:



- 7.50.1.17** `int ipsec4_get_policy (struct inpcb *inp, caddr_t request, size_t len, struct mbuf **mp)`

Definition at line 1480 of file ipsec.c.

References IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_get_policy(), and ipseclog.

Here is the call graph for this function:



7.50.1.18 `static void ipsec4_get_ulp (struct mbuf * m, struct secpolicyindex * spidx, int needport)`
 [static]

Definition at line 950 of file ipsec.c.

References secpolicyindex::dst, IPSEC_PORT_ANY, IPSEC_ULPROTO_ANY, secpolicyindex::src, and secpolicyindex::ul_proto.

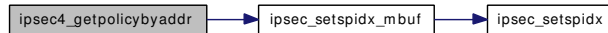
7.50.1.19 `struct secpolicy* ipsec4_getpolicybyaddr (struct mbuf * m, u_int dir, int flag, int * error)`

Definition at line 560 of file ipsec.c.

References ip4_def_policy, ipsec_setspidx_mbuf(), KEYDEBUG, and secpolicy::refcnt.

Referenced by ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:



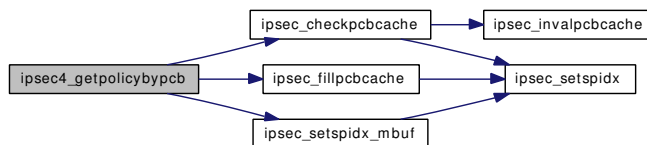
7.50.1.20 `struct secpolicy* ipsec4_getpolicybypcb (struct mbuf * m, u_int dir, struct inpcb * inp, int * error)`

Definition at line 414 of file ipsec.c.

References ip4_def_policy, ipsec_checkpcbcache(), IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_fillpcbcache(), IPSEC_POLICY_BYPASS, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, ipsec_setspidx_mbuf(), ipseclog, ipsecstat, KEYDEBUG, secpolicy::policy, inpcbpolicy::priv, secpolicy::refcnt, inpcbpolicy::sp_in, inpcbpolicy::sp_out, ipsecstat::spdcachelookup, and ipsecstat::spdcachemiss.

Referenced by ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:

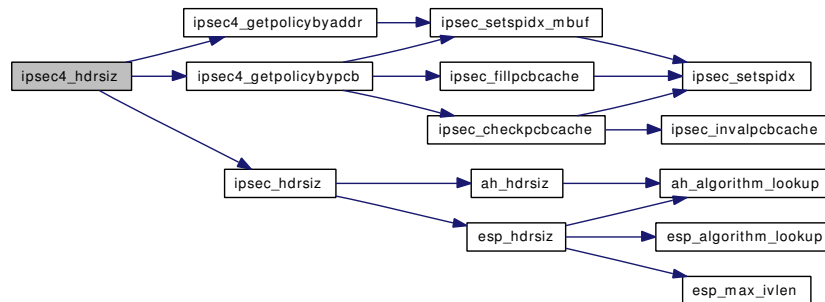


7.50.1.21 `size_t ipsec4_hdrsiz (struct mbuf * m, u_int dir, struct inpcb * inp)`

Definition at line 1976 of file ipsec.c.

References `ipsec4_getpolicybyaddr()`, `ipsec4_getpolicybypcb()`, `ipsec_hdrsiz()`, and `KEYDEBUG`.

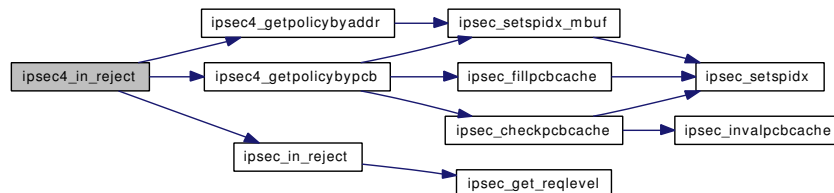
Here is the call graph for this function:

**7.50.1.22** `int ipsec4_in_reject (struct mbuf * m, struct inpcb * inp)`

Definition at line 1823 of file ipsec.c.

References `ipsec4_getpolicybyaddr()`, `ipsec4_getpolicybypcb()`, `IPSEC_DIR_INBOUND`, `ipsec_in_reject()`, and `KEYDEBUG`.

Here is the call graph for this function:

**7.50.1.23** `const char* ipsec4_logpacketstr (struct ip * ip, u_int32_t spi)`

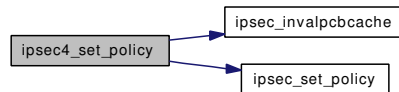
Definition at line 2462 of file ipsec.c.

7.50.1.24 `int ipsec4_set_policy (struct inpcb * inp, int optname, caddr_t request, size_t len, int priv)`

Definition at line 1444 of file ipsec.c.

References `IPSEC_DIR_ANY`, `IPSEC_DIR_INBOUND`, `IPSEC_DIR_OUTBOUND`, `ipsec_invalpcbcache()`, `ipsec_set_policy()`, and `ipseclog`.

Here is the call graph for this function:



7.50.1.25 `static int ipsec4_setspidx_ipaddr (struct mbuf * m, struct secpolicyindex * spidx)`
`[static]`

Definition at line 1028 of file ipsec.c.

References secpolicyindex::dst, secpolicyindex::prefd, secpolicyindex::prefs, and secpolicyindex::src.

7.50.1.26 `int ipsec4_tunnel_validate (struct mbuf * m, int off, u_int nxt0, struct secasvar * sav)`

Definition at line 3297 of file ipsec.c.

References IPSEC_MODE_TRANSPORT.

7.50.1.27 `static struct ipsecaux* ipsec_addaux (struct mbuf * m)` `[static]`

Definition at line 3558 of file ipsec.c.

Referenced by ipsec_addhist().

7.50.1.28 `int ipsec_addhist (struct mbuf * m, int proto, u_int32_t spi)`

Definition at line 3611 of file ipsec.c.

References ipsecaux::hdrs, and ipsec_addaux().

Here is the call graph for this function:



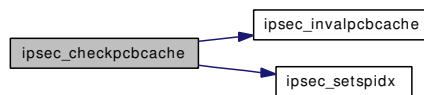
7.50.1.29 `static struct secpolicy* ipsec_checkpcbcache (struct mbuf * m, struct inpcbpolicy * pcbsp, int dir)` `[static]`

Definition at line 250 of file ipsec.c.

References inpcbpolicy::cache, inpcbpolicy::cacheflags, inpcbpolicy::cachegen, inpcbpolicy::cacheidx, IPSEC_DIR_ANY, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_invalpcbcache(), IPSEC_PCbsp_CONNECTED, ipsec_setspidx(), IPSEC_SPSTATE_ALIVE, KEYDEBUG, secpolicy::lastused, secpolicy::refcnt, sp_cachegen, secpolicy::spidx, and secpolicy::state.

Referenced by ipsec4_getpolicybypcb().

Here is the call graph for this function:



7.50.1.30 int ipsec_chkreply (u_int32_t seq, struct secasvar * sav)

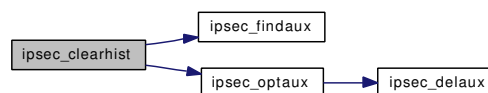
Definition at line 2282 of file ipsec.c.

7.50.1.31 void ipsec_clearhist (struct mbuf * m)

Definition at line 3638 of file ipsec.c.

References ipsec_findaux(), and ipsec_optaux().

Here is the call graph for this function:



7.50.1.32 int ipsec_copy_pcbpolicy (struct inpcbpolicy * old, struct inpcbpolicy * new)

Definition at line 1249 of file ipsec.c.

References ipsec_deepcopy_policy(), and IPSEC_POLICY_IPSEC.

Here is the call graph for this function:



7.50.1.33 struct mbuf* ipsec_copypkt (struct mbuf * m)

Definition at line 3463 of file ipsec.c.

Referenced by esp_output().

7.50.1.34 static struct secpolicy* ipsec_deepcopy_policy (struct secpolicy * src) [static]

Definition at line 1303 of file ipsec.c.

References secpolicy::dir, ipsecrequest::level, ipsecrequest::next, secpolicy::policy, secpolicy::req, ipsecrequest::saidx, secpolicy::so, secpolicy::spidx, and secpolicy::state.

Referenced by ipsec_copy_pcbpolicy().

7.50.1.35 void ipsec_delaux (struct mbuf * m)

Definition at line 3588 of file ipsec.c.

Referenced by ip6_output(), and ipsec_optaux().

7.50.1.36 static void ipsec_delpcbpolicy (struct [inpcbpolicy](#) * p) [static]

Definition at line 1171 of file ipsec.c.

Referenced by ipsec4_delete_pcbpolicy().

7.50.1.37 void ipsec_dumpmbuf (struct mbuf * m)

Definition at line 2566 of file ipsec.c.

7.50.1.38 static int ipsec_fillpcbcache (struct [inpcbpolicy](#) * pcbsp, struct mbuf * m, struct [secpolicy](#) * sp, int dir) [static]

Definition at line 317 of file ipsec.c.

References IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_setspidx(), KEYDEBUG, secpolicy::refcnt, and sp_cachegen.

Referenced by ipsec4_getpolicybypcb().

Here is the call graph for this function:

**7.50.1.39 static struct [ipsecaux](#)* ipsec_findaux (struct mbuf * m) [static]**

Definition at line 3578 of file ipsec.c.

Referenced by ipsec_clearhist(), and ipsec_getnhist().

7.50.1.40 static int ipsec_get_policy (struct [secpolicy](#) * sp, struct mbuf ** mp) [static]

Definition at line 1420 of file ipsec.c.

References ipseclog, and KEYDEBUG.

Referenced by ipsec4_get_policy().

7.50.1.41 u_int ipsec_get_reqlevel (struct [ipsecrequest](#) * isr, int af)

Definition at line 1653 of file ipsec.c.

References ip4_ah_net_deflev, ip4_ah_trans_deflev, ip4_esp_net_deflev, ip4_esp_trans_deflev, ip6_ah_net_deflev, ip6_ah_trans_deflev, ip6_esp_net_deflev, ip6_esp_trans_deflev, IPSEC_LEVEL_DEFAULT, IPSEC_LEVEL_REQUIRE, IPSEC_LEVEL_UNIQUE, IPSEC_LEVEL_USE, and IPSEC_MODE_TUNNEL.

Referenced by ipsec_in_reject().

7.50.1.42 int ipsec_getnhist (struct mbuf * m)

Definition at line 3626 of file ipsec.c.

References ipsecaux::hdrs, and ipsec_findaux().

Here is the call graph for this function:



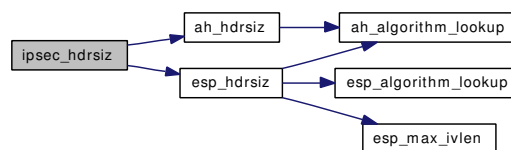
7.50.1.43 static size_t ipsec_hdrsiz (struct secpolicy * sp) [static]

Definition at line 1904 of file ipsec.c.

References ah_hdrsiz(), esp_hdrsiz(), IPSEC_MODE_TUNNEL, IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, ipseclog, KEYDEBUG, ipsecrequest::next, and ipsecrequest::saidx.

Referenced by ipsec4_hdrsiz().

Here is the call graph for this function:



7.50.1.44 static int ipsec_in_reject (struct secpolicy * sp, struct mbuf * m) [static]

Definition at line 1740 of file ipsec.c.

References ipsec_get_reqlevel(), IPSEC_LEVEL_REQUIRE, IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, KEYDEBUG, ipsecrequest::level, M_AUTHIPDGM, M_AUTHIPHDR, M_DECRYPTED, ipsecrequest::next, ipsecrequest::saidx, and ipsecrequest::sav.

Referenced by ipsec4_in_reject().

Here is the call graph for this function:



7.50.1.45 int ipsec_init_pcbpolicy (struct socket * so, struct inpcbpolicy ** pcb_sp)

Definition at line 1180 of file ipsec.c.

References `secpolicy::dir`, `IPSEC_DIR_INBOUND`, `IPSEC_DIR_OUTBOUND`, `ipsec_newpcbpolicy()`, `IPSEC_POLICY_ENTRUST`, `IPSEC_SPSTATE_ALIVE`, `ipseclog`, `secpolicy::persist`, `secpolicy::policy`, `secpolicy::readonly`, `secpolicy::refcnt`, `secpolicy::so`, and `secpolicy::state`.

Here is the call graph for this function:



7.50.1.46 `static int ipsec_invalpcbcache (struct inpcbpolicy * pcbsp, int dir)` [static]

Definition at line 355 of file `ipsec.c`.

References `IPSEC_DIR_ANY`, `IPSEC_DIR_INBOUND`, and `IPSEC_DIR_OUTBOUND`.

Referenced by `ipsec4_delete_pcbpolicy()`, `ipsec4_set_policy()`, `ipsec_checkpcbcache()`, `ipsec_pcbconn()`, and `ipsec_pcbdisconn()`.

7.50.1.47 `int ipsec_invalpcbcacheall ()`

Definition at line 394 of file `ipsec.c`.

References `sp_cachegen`.

7.50.1.48 `const char* ipsec_logsastr (struct secasvar * sav)`

Definition at line 2519 of file `ipsec.c`.

References `INET6_ADDRSTRLEN`, and `ip6_sprintf()`.

Referenced by `ipsec_updatereplay()`.

Here is the call graph for this function:



7.50.1.49 `static struct inpcbpolicy* ipsec_newpcbpolicy ()` [static]

Definition at line 1162 of file `ipsec.c`.

Referenced by `ipsec_init_pcbpolicy()`.

7.50.1.50 `static void ipsec_optaux (struct mbuf * m, struct ipsecaux * aux)` [static]

Definition at line 3600 of file `ipsec.c`.

References `ipsec_delaux()`.

Referenced by `ipsec_clearhist()`.

Here is the call graph for this function:



7.50.1.51 int ipsec_pcbconn (struct [inpcbpolicy](#) * *pcb*)

Definition at line 374 of file ipsec.c.

References IPSEC_DIR_ANY, ipsec_invalpcbcache(), and IPSEC_PCbsp_CONNECTED.

Referenced by in6_pcbconnect().

Here is the call graph for this function:



7.50.1.52 int ipsec_pcbdisconn (struct [inpcbpolicy](#) * *pcb*)

Definition at line 384 of file ipsec.c.

References IPSEC_DIR_ANY, ipsec_invalpcbcache(), and IPSEC_PCbsp_CONNECTED.

Referenced by in6_pcbdisconnect().

Here is the call graph for this function:



7.50.1.53 static int ipsec_set_policy (struct [secpolicy](#) ** *spp*, int *optname*, [caddr_t](#) *request*, [size_t](#) *len*, int *priv*) [static]

Definition at line 1371 of file ipsec.c.

References IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_NONE, IPSEC_SPSTATE_ALIVE, KEYDEBUG, and secpolicy::state.

Referenced by ipsec4_set_policy().

7.50.1.54 static int ipsec_setspidx (struct [mbuf](#) * *m*, struct [secpolicyindex](#) * *spidx*, int *needport*) [static]

Definition at line 866 of file ipsec.c.

References KEYDEBUG.

Referenced by ipsec_checkpcbcache(), ipsec_fillpcbcache(), and ipsec_setspidx_mbuf().

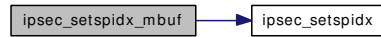
7.50.1.55 int ipsec_setspidx_mbuf (struct [secpolicyindex](#) * *spidx*, int *family*, struct [mbuf](#) * *m*, int *needport*)

Definition at line 834 of file ipsec.c.

References ipsec_setspidx().

Referenced by ipsec4_getpolicybyaddr(), and ipsec4_getpolicybypcb().

Here is the call graph for this function:

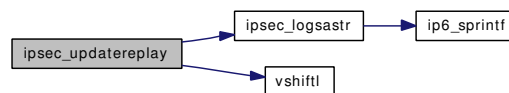


7.50.1.56 int ipsec_updatereplay (u_int32_t seq, struct secasvar * sav)

Definition at line 2342 of file ipsec.c.

References ipsec_logsastr(), ipseclog, and vshiftl().

Here is the call graph for this function:



- 7.50.1.57 `NET_NEEDS_GIANT ("ipsec")`
- 7.50.1.58 `SYSCTL_DECL (_net_inet_ipsec)`
- 7.50.1.59 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ESP_RANDPAD, esp_randpad, CTLFLAG_RW, & ip4_esp_randpad, 0, "")`
- 7.50.1.60 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEBUG, debug, CTLFLAG_RW, & ipsec_debug, 0, "")`
- 7.50.1.61 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ECN, ecn, CTLFLAG_RW, & ip4_ipsec_ecn, 0, "")`
- 7.50.1.62 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DFBIT, dfbit, CTLFLAG_RW, & ip4_ipsec_dfbit, 0, "")`
- 7.50.1.63 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_OFFSETMASK, ah_offsetmask, CTLFLAG_RW, & ip4_ah_offsetmask, 0, "")`
- 7.50.1.64 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_CLEARRTOS, ah_clearartos, CTLFLAG_RW, & ip4_ah_clearartos, 0, "")`
- 7.50.1.65 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_NETLEV, ah_net_deflev, CTLFLAG_RW, & ip4_ah_net_deflev, 0, "")`
- 7.50.1.66 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_TRANSLEV, ah_trans_deflev, CTLFLAG_RW, & ip4_ah_trans_deflev, 0, "")`
- 7.50.1.67 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_NETLEV, esp_net_deflev, CTLFLAG_RW, & ip4_esp_net_deflev, 0, "")`
- 7.50.1.68 `SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_TRANSLEV, esp_trans_deflev, CTLFLAG_RW, & ip4_esp_trans_deflev, 0, "")`
- 7.50.1.69 `SYSCTL_STRUCT (_net_inet_ipsec, IPSECCTL_STATS, stats, CTLFLAG_RD, & ipsecstat, ipsecstat, "")`
- 7.50.1.70 `static void vshiftl (unsigned char * bitmap, int nbit, int wsize) [static]`

Definition at line 2441 of file ipsec.c.

Referenced by ipsec_updatereplay().

7.50.2 Variable Documentation

7.50.2.1 `int ip4_ah_clearartos = 1`

Definition at line 117 of file ipsec.c.

7.50.2.2 `int ip4_ah_net_deflev = IPSEC_LEVEL_USE`

Definition at line 123 of file ipsec.c.

Referenced by `ipsec_get_reqlevel()`.

7.50.2.3 `int ip4_ah_offsetmask = 0`

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

7.50.2.4 `int ip4_ah_trans_deflev = IPSEC_LEVEL_USE`

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

Referenced by `ipsec_get_reqlevel()`.

7.50.2.5 `struct secpolicy* ip4_def_policy`

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

Referenced by `ipsec4_getpolicybyaddr()`, and `ipsec4_getpolicybypcb()`.

7.50.2.6 `int ip4_esp_net_deflev = IPSEC_LEVEL_USE`

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

Referenced by `ipsec_get_reqlevel()`.

7.50.2.7 `int ip4_esp_randpad = -1`

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

7.50.2.8 `int ip4_esp_trans_deflev = IPSEC_LEVEL_USE`

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

Referenced by `ipsec_get_reqlevel()`.

7.50.2.9 `int ip4_ipsec_dfbit = 0`

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

7.50.2.10 `int ip4_ipsec_ecn = 0`

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

7.50.2.11 `int ipsec_debug = 0`

Definition at line 111 of file `ipsec.c`.

7.50.2.12 struct ipsecstat ipsecstat

Definition at line 116 of file ipsec.c.

Referenced by ipsec4_getpolicybypcb().

7.50.2.13 int sp_cachegen = 1 [static]

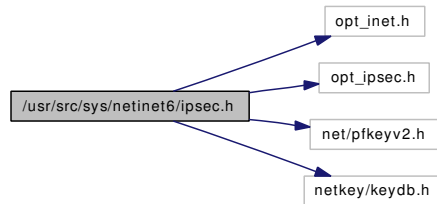
Definition at line 128 of file ipsec.c.

Referenced by ipsec_checkpcbcache(), ipsec_fillpcbcache(), and ipsec_invalpcbcacheall().

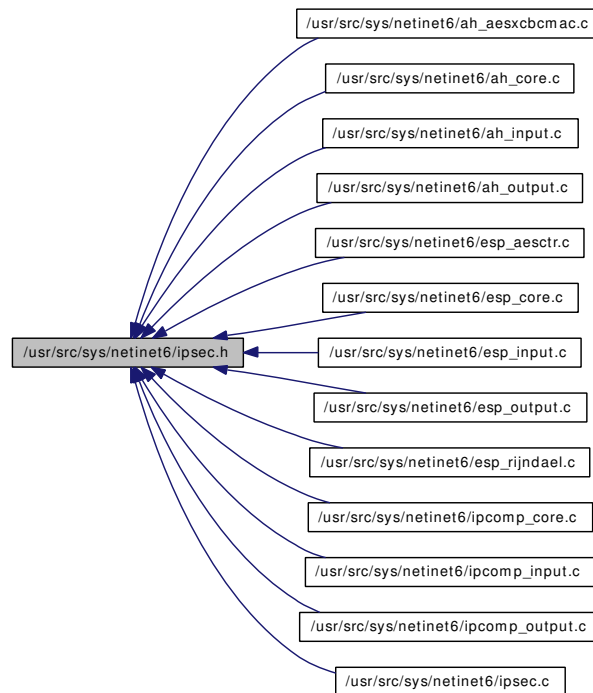
7.51 /usr/src/sys/netinet6/ipsec.h File Reference

```
#include "opt_inet.h"
#include "opt_ipsec.h"
#include <net/pfkeyv2.h>
#include <netkey/keydb.h>
```

Include dependency graph for ipsec.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [secpolicyindex](#)
- struct [secpolicy](#)
- struct [ipsecrequest](#)
- struct [inpcbpolicy](#)
- struct [secspacq](#)

- struct [ipsecaux](#)
- struct [ipsecstat](#)
- struct [ipsec_output_state](#)
- struct [ipsec_history](#)

Defines

- #define [IPSEC_MANUAL_POLICYID_MAX](#) 0x3fff
- #define [IPSEC_SPSTATE_DEAD](#) 0
- #define [IPSEC_SPSTATE_ALIVE](#) 1
- #define [IPSEC_PCBSP_CONNECTED](#) 1
- #define [IPSEC_PORT_ANY](#) 0
- #define [IPSEC_ULPROTO_ANY](#) 255
- #define [IPSEC_PROTO_ANY](#) 255
- #define [IPSEC_MODE_ANY](#) 0
- #define [IPSEC_MODE_TRANSPORT](#) 1
- #define [IPSEC_MODE_TUNNEL](#) 2
- #define [IPSEC_MODE_TCPMD5](#) 3
- #define [IPSEC_DIR_ANY](#) 0
- #define [IPSEC_DIR_INBOUND](#) 1
- #define [IPSEC_DIR_OUTBOUND](#) 2
- #define [IPSEC_DIR_MAX](#) 3
- #define [IPSEC_DIR_INVALID](#) 4
- #define [IPSEC_POLICY_DISCARD](#) 0
- #define [IPSEC_POLICY_NONE](#) 1
- #define [IPSEC_POLICY_IPSEC](#) 2
- #define [IPSEC_POLICY_ENTRUST](#) 3
- #define [IPSEC_POLICY_BYPASS](#) 4
- #define [IPSEC_POLICY_TCP](#) 5
- #define [IPSEC_LEVEL_DEFAULT](#) 0
- #define [IPSEC_LEVEL_USE](#) 1
- #define [IPSEC_LEVEL_REQUIRE](#) 2
- #define [IPSEC_LEVEL_UNIQUE](#) 3
- #define [IPSEC_MANUAL_REQID_MAX](#) 0x3fff
- #define [IPSEC_REPLAYWSIZE](#) 32
- #define [IPSECCTL_STATS](#) 1
- #define [IPSECCTL_DEF_POLICY](#) 2
- #define [IPSECCTL_DEF_ESP_TRANSLEV](#) 3
- #define [IPSECCTL_DEF_ESP_NETLEV](#) 4
- #define [IPSECCTL_DEF_AH_TRANSLEV](#) 5
- #define [IPSECCTL_DEF_AH_NETLEV](#) 6
- #define [IPSECCTL_AH_CLEAROS](#) 8
- #define [IPSECCTL_AH_OFFSETMASK](#) 9
- #define [IPSECCTL_DFBIT](#) 10
- #define [IPSECCTL_ECN](#) 11
- #define [IPSECCTL_DEBUG](#) 12
- #define [IPSECCTL_ESP_RANDPAD](#) 13
- #define [IPSECCTL_MAXID](#) 14
- #define [IPSECCTL_NAMES](#)
- #define [IPSEC6CTL_NAMES](#)
- #define [ipseclog\(x\)](#) do { if ([ipsec_debug](#)) log x; } while (/*CONSTCOND*/ 0)

Functions

- int ipsec_pcbconn __P ((struct inpcbpolicy *))
- int ipsec_invalpcbcacheall __P ((void))
- secpolicy *ipsec4_getpolicybypcb __P ((struct mbuf *, u_int, struct inpcb *, int *))
- secpolicy *ipsec4_getpolicybyaddr __P ((struct mbuf *, u_int, int, int *))
- int ipsec_init_pcbpolicy __P ((struct socket *, struct inpcbpolicy **))
- int ipsec_copy_pcbpolicy __P ((struct inpcbpolicy *, struct inpcbpolicy *))
- u_int ipsec_get_reqlevel __P ((struct ipsecrequest *, int))
- int ipsec4_set_policy __P ((struct inpcb *, int, caddr_t, size_t, int))
- int ipsec4_get_policy __P ((struct inpcb *, caddr_t, size_t, struct mbuf **))
- int ipsec4_delete_pcbpolicy __P ((struct inpcb *))
- int ipsec4_in_reject __P ((struct mbuf *, struct inpcb *))
- int ipsec_chkreplay __P ((u_int32_t, struct secasvar *))
- size_t ipsec4_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))
- size_t ipsec_hdrsiz_tcp __P ((struct tcpcb *))
- const char *ipsec4_logpacketstr __P ((struct ip *, u_int32_t))
- const char *ipsec_logsastr __P ((struct secasvar *))
- void ipsec_dumpmbuf __P ((struct mbuf *))
- int ipsec4_output __P ((struct ipsec_output_state *, struct secpolicy *, int))
- int ipsec4_tunnel_validate __P ((struct mbuf *, int, u_int, struct secasvar *))
- int ipsec_addhist __P ((struct mbuf *, int, u_int32_t))

Variables

- int ipsec_debug

7.51.1 Define Documentation

7.51.1.1 #define IPSEC6CTL_NAMES

Value:

```
{ \
    { 0, 0 }, \
    { 0, 0 }, \
    { "def_policy", CTLTYPE_INT }, \
    { "esp_trans_deflev", CTLTYPE_INT }, \
    { "esp_net_deflev", CTLTYPE_INT }, \
    { "ah_trans_deflev", CTLTYPE_INT }, \
    { "ah_net_deflev", CTLTYPE_INT }, \
    { 0, 0 }, \
    { 0, 0 }, \
    { 0, 0 }, \
    { 0, 0 }, \
    { "ecn", CTLTYPE_INT }, \
    { "debug", CTLTYPE_INT }, \
    { "esp_randpad", CTLTYPE_INT }, \
}
```

Definition at line 284 of file ipsec.h.

7.51.1.2 #define IPSEC_DIR_ANY 0

Definition at line 173 of file ipsec.h.

Referenced by ipsec4_delete_pcbpolicy(), ipsec4_set_policy(), ipsec_checkpcbcache(), ipsec_invalpcbcache(), ipsec_pcbconn(), and ipsec_pcbdisconn().

7.51.1.3 #define IPSEC_DIR_INBOUND 1

Definition at line 174 of file ipsec.h.

Referenced by ipsec4_get_policy(), ipsec4_getpolicybypcb(), ipsec4_in_reject(), ipsec4_set_policy(), ipsec_checkpcbcache(), ipsec_fillpcbcache(), ipsec_init_pcbpolicy(), and ipsec_invalpcbcache().

7.51.1.4 #define IPSEC_DIR_INVALID 4

Definition at line 177 of file ipsec.h.

7.51.1.5 #define IPSEC_DIR_MAX 3

Definition at line 176 of file ipsec.h.

7.51.1.6 #define IPSEC_DIR_OUTBOUND 2

Definition at line 175 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec4_get_policy(), ipsec4_getpolicybypcb(), ipsec4_set_policy(), ipsec_checkpcbcache(), ipsec_fillpcbcache(), ipsec_init_pcbpolicy(), and ipsec_invalpcbcache().

7.51.1.7 #define IPSEC_LEVEL_DEFAULT 0

Definition at line 193 of file ipsec.h.

Referenced by ipsec_get_reqlevel().

7.51.1.8 #define IPSEC_LEVEL_REQUIRE 2

Definition at line 195 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), and ipsec_in_reject().

7.51.1.9 #define IPSEC_LEVEL_UNIQUE 3

Definition at line 196 of file ipsec.h.

Referenced by ipsec_get_reqlevel().

7.51.1.10 #define IPSEC_LEVEL_USE 1

Definition at line 194 of file ipsec.h.

Referenced by ipsec_get_reqlevel().

7.51.1.11 #define IPSEC_MANUAL_POLICYID_MAX 0x3fff

Definition at line 81 of file ipsec.h.

7.51.1.12 #define IPSEC_MANUAL_REQID_MAX 0x3fff

Definition at line 198 of file ipsec.h.

7.51.1.13 #define IPSEC_MODE_ANY 0

Definition at line 163 of file ipsec.h.

Referenced by ip6_forward().

7.51.1.14 #define IPSEC_MODE_TCPMD5 3

Definition at line 166 of file ipsec.h.

7.51.1.15 #define IPSEC_MODE_TRANSPORT 1

Definition at line 164 of file ipsec.h.

Referenced by ipsec4_tunnel_validate().

7.51.1.16 #define IPSEC_MODE_TUNNEL 2

Definition at line 165 of file ipsec.h.

Referenced by ip6_forward(), ipsec_get_reqlevel(), and ipsec_hdrsiz().

7.51.1.17 #define IPSEC_PCBSP_CONNECTED 1

Definition at line 137 of file ipsec.h.

Referenced by ipsec_checkpcbcache(), ipsec_pcbconn(), and ipsec_pcbdisconn().

7.51.1.18 #define IPSEC_POLICY_BYPASS 4

Definition at line 189 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec4_getpolicybypcb(), ipsec_hdrsiz(), ipsec_in_reject(), and ipsec_set_policy().

7.51.1.19 #define IPSEC_POLICY_DISCARD 0

Definition at line 185 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec_hdrsiz(), ipsec_in_reject(), and ipsec_set_policy().

7.51.1.20 #define IPSEC_POLICY_ENTRUST 3

Definition at line 188 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec4_getpolicybypcb(), ipsec_hdrsiz(), ipsec_in_reject(), and ipsec_init_pcbpolicy().

7.51.1.21 #define IPSEC_POLICY_IPSEC 2

Definition at line 187 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec4_getpolicybypcb(), ipsec_copy_pcbpolicy(), ipsec_hdrsiz(), and ipsec_in_reject().

7.51.1.22 #define IPSEC_POLICY_NONE 1

Definition at line 186 of file ipsec.h.

Referenced by ip6_forward(), ip6_output(), ipsec_hdrsiz(), ipsec_in_reject(), and ipsec_set_policy().

7.51.1.23 #define IPSEC_POLICY_TCP 5

Definition at line 190 of file ipsec.h.

7.51.1.24 #define IPSEC_PORT_ANY 0

Definition at line 157 of file ipsec.h.

Referenced by ipsec4_get_ulp().

7.51.1.25 #define IPSEC_PROTO_ANY 255

Definition at line 159 of file ipsec.h.

7.51.1.26 #define IPSEC_REPLAYWSIZE 32

Definition at line 209 of file ipsec.h.

7.51.1.27 #define IPSEC_SPSTATE_ALIVE 1

Definition at line 89 of file ipsec.h.

Referenced by ipsec_checkpcbcache(), ipsec_init_pcbpolicy(), and ipsec_set_policy().

7.51.1.28 #define IPSEC_SPSTATE_DEAD 0

Definition at line 88 of file ipsec.h.

7.51.1.29 #define IPSEC_ULPROTO_ANY 255

Definition at line 158 of file ipsec.h.

Referenced by ipsec4_get_ulp().

7.51.1.30 #define IPSECCTL_AH_CLEARRTOS 8

Definition at line 259 of file ipsec.h.

7.51.1.31 #define IPSECCTL_AH_OFFSETMASK 9

Definition at line 260 of file ipsec.h.

7.51.1.32 #define IPSECCTL_DEBUG 12

Definition at line 263 of file ipsec.h.

7.51.1.33 #define IPSECCTL_DEF_AH_NETLEV 6

Definition at line 255 of file ipsec.h.

7.51.1.34 #define IPSECCTL_DEF_AH_TRANSLEV 5

Definition at line 254 of file ipsec.h.

7.51.1.35 #define IPSECCTL_DEF_ESP_NETLEV 4

Definition at line 253 of file ipsec.h.

7.51.1.36 #define IPSECCTL_DEF_ESP_TRANSLEV 3

Definition at line 252 of file ipsec.h.

7.51.1.37 #define IPSECCTL_DEF_POLICY 2

Definition at line 251 of file ipsec.h.

7.51.1.38 #define IPSECCTL_DFBIT 10

Definition at line 261 of file ipsec.h.

7.51.1.39 #define IPSECCTL_ECN 11

Definition at line 262 of file ipsec.h.

7.51.1.40 #define IPSECCTL_ESP_RANDPAD 13

Definition at line 264 of file ipsec.h.

7.51.1.41 #define IPSECCTL_MAXID 14

Definition at line 265 of file ipsec.h.

7.51.1.42 #define IPSECCTL_NAMES

Value:

```
{ \
    { 0, 0 }, \
    { 0, 0 }, \
    { "def_policy", CTLTYPE_INT }, \
    { "esp_trans_deflev", CTLTYPE_INT }, \
    { "esp_net_deflev", CTLTYPE_INT }, \
    { "ah_trans_deflev", CTLTYPE_INT }, \
    { "ah_net_deflev", CTLTYPE_INT }, \
    { 0, 0 }, \
    { "ah_clearartos", CTLTYPE_INT }, \
    { "ah_offsetmask", CTLTYPE_INT }, \
    { "dfbit", CTLTYPE_INT }, \
    { "ecn", CTLTYPE_INT }, \
    { "debug", CTLTYPE_INT }, \
    { "esp_randpad", CTLTYPE_INT }, \
}
```

Definition at line 267 of file ipsec.h.

7.51.1.43 #define IPSECCTL_STATS 1

Definition at line 250 of file ipsec.h.

7.51.1.44 #define ipseclog(x) do { if (ipsec_debug) log x; } while (/*CONSTCOND*/ 0)

Definition at line 330 of file ipsec.h.

Referenced by ah_common_mature(), ah_none_mature(), deflate_common(), esp_aesctr_decrypt(), esp_aesctr_encrypt(), esp_aesctr_mature(), esp_auth(), esp_cbc_decrypt(), esp_cbc_encrypt(), esp_cbc_mature(), esp_descbc_mature(), esp_output(), esp_schedule(), ipcomp_output(), ipsec4_get_policy(), ipsec4_getpolicybypcb(), ipsec4_set_policy(), ipsec_get_policy(), ipsec_hdrsiz(), ipsec_init_pcbpolicy(), and ipsec_updatereplay().

7.51.2 Function Documentation

- 7.51.2.1 `int ipsec_addhist __P ((struct mbuf *, int, u_int32_t))`
- 7.51.2.2 `int ipsec4_tunnel_validate __P ((struct mbuf *, int, u_int, struct secasvar *))`
- 7.51.2.3 `int ipsec4_output __P ((struct ipsec_output_state *, struct secpolicy *, int))`
- 7.51.2.4 `void ipsec_dumpmbuf __P ((struct mbuf *))`
- 7.51.2.5 `const char* ipsec_logsastr __P ((struct secasvar *))`
- 7.51.2.6 `const char* ipsec4_logpacketstr __P ((struct ip *, u_int32_t))`
- 7.51.2.7 `size_t ipsec_hdrsiz_tcp __P ((struct tcpcb *))`
- 7.51.2.8 `size_t ipsec4_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))`
- 7.51.2.9 `int ipsec_updatereplay __P ((u_int32_t, struct secasvar *))`
- 7.51.2.10 `int ipsec4_in_reject __P ((struct mbuf *, struct inpcb *))`
- 7.51.2.11 `int ipsec4_delete_pcbpolicy __P ((struct inpcb *))`
- 7.51.2.12 `int ipsec4_get_policy __P ((struct inpcb *, caddr_t, size_t, struct mbuf **))`
- 7.51.2.13 `int ipsec4_set_policy __P ((struct inpcb *, int, caddr_t, size_t, int))`
- 7.51.2.14 `u_int ipsec_get_reqlevel __P ((struct ipsecrequest *, int))`
- 7.51.2.15 `int ipsec_copy_pcbpolicy __P ((struct inpcbpolicy *, struct inpcbpolicy *))`
- 7.51.2.16 `int ipsec_init_pcbpolicy __P ((struct socket *, struct inpcbpolicy **))`
- 7.51.2.17 `struct secpolicy* ipsec4_getpolicybyaddr __P ((struct mbuf *, u_int, int, int *))`
- 7.51.2.18 `struct secpolicy* ipsec4_getpolicybypcb __P ((struct mbuf *, u_int, struct inpcb *, int *))`
- 7.51.2.19 `int ipsec_invalpcbcacheall __P ((void))`
- 7.51.2.20 `int ipsec_pcbdisconn __P ((struct inpcbpolicy *))`

7.51.3 Variable Documentation

- 7.51.3.1 `int ipsec_debug`

Definition at line 111 of file ipsec.c.

7.52 /usr/src/sys/netinet6/ipsec6.h File Reference

```
#include <net/pfkeyv2.h>
```

```
#include <netkey/keydb.h>
```

Include dependency graph for ipsec6.h:



Functions

- [secpolicy](#) *ipsec6_getpolicybypcb [__P](#) ((struct mbuf *, u_int, struct inpcb *, int *))
- [secpolicy](#) *ipsec6_getpolicybyaddr [__P](#) ((struct mbuf *, u_int, int, int *))
- int ipsec6_delete_pcbpolicy [__P](#) ((struct inpcb *))
- int ipsec6_set_policy [__P](#) ((struct inpcb *, int, caddr_t, [size_t](#), int))
- int ipsec6_get_policy [__P](#) ((struct inpcb *, caddr_t, [size_t](#), struct mbuf **))
- int ipsec6_in_reject [__P](#) ((struct mbuf *, struct inpcb *))
- [size_t](#) ipsec6_hdrsiz [__P](#) ((struct mbuf *, u_int, struct inpcb *))
- const char *ipsec6_logpacketstr [__P](#) ((struct ip6_hdr *, u_int32_t))
- int ipsec6_output_trans [__P](#) ((struct [ipsec_output_state](#) *, u_char *, struct mbuf *, struct [secpolicy](#) *, int, int *))
- int ipsec6_output_tunnel [__P](#) ((struct [ipsec_output_state](#) *, struct [secpolicy](#) *, int))
- int ipsec6_tunnel_validate [__P](#) ((struct mbuf *, int, u_int, struct secasvar *))

Variables

- [ipsecstat](#) ipsec6stat
- [secpolicy](#) * ip6_def_policy
- int ip6_esp_trans_deflev
- int ip6_esp_net_deflev
- int ip6_ah_trans_deflev
- int ip6_ah_net_deflev
- int ip6_ipsec_ecn
- int ip6_esp_randpad

7.52.1 Function Documentation

- 7.52.1.1 `int ipsec6_tunnel_validate __P ((struct mbuf *, int, u_int, struct secasvar *))`
- 7.52.1.2 `int ipsec6_output_tunnel __P ((struct ipsec_output_state *, struct secpolicy *, int))`
- 7.52.1.3 `int ipsec6_output_trans __P ((struct ipsec_output_state *, u_char *, struct mbuf *, struct secpolicy *, int, int *))`
- 7.52.1.4 `const char* ipsec6_logpacketstr __P ((struct ip6_hdr *, u_int32_t))`
- 7.52.1.5 `size_t ipsec6_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))`
- 7.52.1.6 `int ipsec6_in_reject __P ((struct mbuf *, struct inpcb *))`
- 7.52.1.7 `int ipsec6_get_policy __P ((struct inpcb *, caddr_t, size_t, struct mbuf **))`
- 7.52.1.8 `int ipsec6_set_policy __P ((struct inpcb *, int, caddr_t, size_t, int))`
- 7.52.1.9 `int ipsec6_delete_pcbpolicy __P ((struct inpcb *))`
- 7.52.1.10 `struct secpolicy* ipsec6_getpolicybyaddr __P ((struct mbuf *, u_int, int, int *))`
- 7.52.1.11 `struct secpolicy* ipsec6_getpolicybypcb __P ((struct mbuf *, u_int, struct inpcb *, int *))`

7.52.2 Variable Documentation

7.52.2.1 `int ip6_ah_net_deflev`

Referenced by `ipsec_get_reqlevel()`.

7.52.2.2 `int ip6_ah_trans_deflev`

Referenced by `ipsec_get_reqlevel()`.

7.52.2.3 `struct secpolicy* ip6_def_policy`

7.52.2.4 `int ip6_esp_net_deflev`

Referenced by `ipsec_get_reqlevel()`.

7.52.2.5 `int ip6_esp_randpad`

7.52.2.6 `int ip6_esp_trans_deflev`

Referenced by `ipsec_get_reqlevel()`.

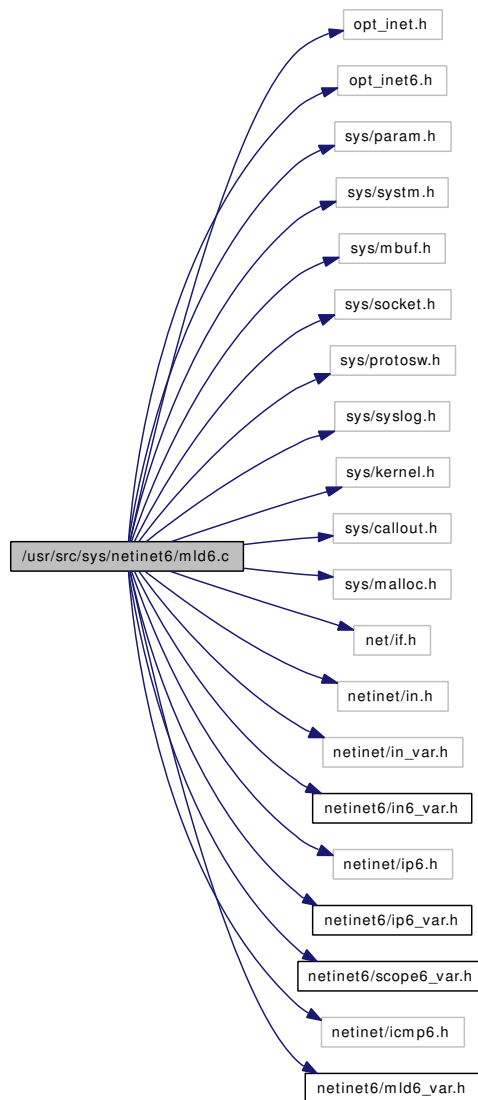
7.52.2.7 int `ip6_ipsec_ecn`**7.52.2.8** struct `ipsestat ipsec6stat`

Referenced by `esp_output()`, `ip6_forward()`, `ip6_input()`, `ip6_output()`, `ipcomp_output()`, `rip6_input()`, `sctp6_input()`, and `udp6_append()`.

7.53 /usr/src/sys/netinet6/mld6.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/system.h>  
#include <sys/mbuf.h>  
#include <sys/socket.h>  
#include <sys/protosw.h>  
#include <sys/syslog.h>  
#include <sys/kernel.h>  
#include <sys/callout.h>  
#include <sys/malloc.h>  
#include <net/if.h>  
#include <netinet/in.h>  
#include <netinet/in_var.h>  
#include <netinet6/in6_var.h>  
#include <netinet/ip6.h>  
#include <netinet6/ip6_var.h>  
#include <netinet6/scope6_var.h>  
#include <netinet/icmp6.h>  
#include <netinet6/mld6_var.h>
```

Include dependency graph for mld6.c:



Defines

- `#define MLD_TIMER_SCALE 1000`
- `#define MLD_UNSOLICITED_REPORT_INTERVAL 10`

Functions

- static void `mld6_sendpkt` (struct `in6_multi *`, int, const struct `in6_addr *`)
- static void `mld_starttimer` (struct `in6_multi *`)
- static void `mld_stoptimer` (struct `in6_multi *`)
- static void `mld_timeo` (struct `in6_multi *`)
- static u_long `mld_timerresid` (struct `in6_multi *`)
- void `mld6_init` ()
- void `mld6_start_listening` (struct `in6_multi *in6m`)
- void `mld6_stop_listening` (struct `in6_multi *in6m`)

- void `mld6_input` (struct mbuf *m, int off)
- `in6_multi * in6_addmulti` (struct `in6_addr *maddr6`, struct ifnet *ifp, int *errorp, int delay)
- void `in6_delmulti` (struct `in6_multi *in6m`)

Variables

- static struct `ip6_pktopts ip6_opts`

7.53.1 Define Documentation

7.53.1.1 #define MLD_TIMER_SCALE 1000

Definition at line 97 of file mld6.c.

Referenced by `mld6_input()`.

7.53.1.2 #define MLD_UNSOLICITED_REPORT_INTERVAL 10

Definition at line 102 of file mld6.c.

Referenced by `mld6_start_listening()`.

7.53.2 Function Documentation

7.53.2.1 struct `in6_multi*` `in6_addmulti` (struct `in6_addr * maddr6`, struct ifnet * `ifp`, int * `errorp`, int `delay`)

Definition at line 547 of file mld6.c.

References `in6_multihead`, `mld6_start_listening()`, `MLD_REPORTPENDING`, and `mld_starttimer()`.

Referenced by `in6_joingroup()`.

Here is the call graph for this function:



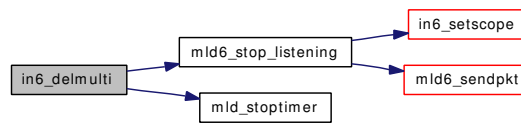
7.53.2.2 void `in6_delmulti` (struct `in6_multi * in6m`)

Definition at line 629 of file mld6.c.

References `mld6_stop_listening()`, and `mld_stoptimer()`.

Referenced by `in6_ifdetach()`, `in6_leavegroup()`, `in6_pcbpurgeif0()`, `in6_purgeaddr()`, `ip6_freemoptions()`, and `ip6_setmoptions()`.

Here is the call graph for this function:



7.53.2.3 void mld6_init (void)

Definition at line 113 of file mld6.c.

References ip6_initpktopts(), ip6_opts, and ip6_pktopts::ip6po_hbh.

Referenced by icmp6_init().

Here is the call graph for this function:



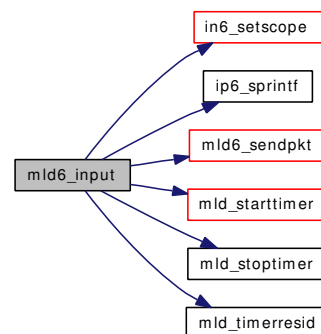
7.53.2.4 void mld6_input (struct mbuf * m, int off)

Definition at line 272 of file mld6.c.

References icmp6stat, IFP_TO_IA6, IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_LINKLOCAL, IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, IN6_LOOKUP_MULTI, in6_setscope(), in6addr_linklocal_allnodes, in6_multi::in6m_addr, in6_multi::in6m_state, in6_multi::in6m_timer, IN6M_TIMER_UNDEF, INET6_ADDRSTRLEN, ip6_sprintf(), IPV6_ADDR_MC_SCOPE, IPV6_ADDR_SCOPE_LINKLOCAL, M_LOOP, mld6_sendpkt(), MLD_IREPORTEDLAST, MLD_OTHERLISTENER, mld_starttimer(), mld_stoptimer(), MLD_TIMER_SCALE, and mld_timerresid().

Referenced by icmp6_input().

Here is the call graph for this function:



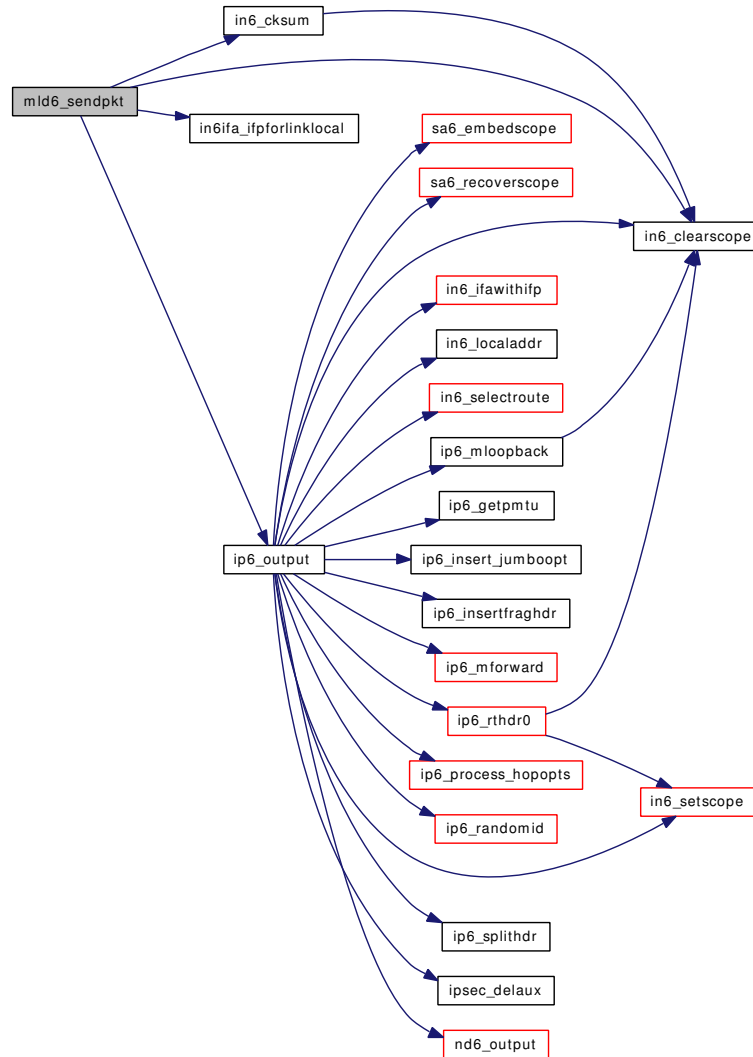
7.53.2.5 static void mld6_sendpkt (struct in6_multi *, int, const struct in6_addr *) [static]

Definition at line 444 of file mld6.c.

References `in6_ifaddr::ia_addr`, `icmp6stat`, `in6_cksum()`, `in6_clearscope()`, `IN6_IFF_ANYCAST`, `IN6_IFF_NOTREADY`, `in6ifa_ifpforlinklocal()`, `ip6_mrouter`, `ip6_opts`, `ip6_output()`, and `sockaddr_in6::sin6_addr`.

Referenced by `mld6_input()`, `mld6_start_listening()`, `mld6_stop_listening()`, and `mld_timeo()`.

Here is the call graph for this function:



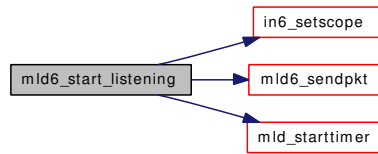
7.53.2.6 void mld6_start_listening (struct `in6_multi` * `in6m`)

Definition at line 212 of file `mld6.c`.

References `IN6_ARE_ADDR_EQUAL`, `in6_setscope()`, `in6addr_linklocal_allnodes`, `IPV6_ADDR_MC_SCOPE`, `IPV6_ADDR_SCOPE_LINKLOCAL`, `mld6_sendpkt()`, `MLD_IREPORTEDLAST`, `MLD_OTHERLISTENER`, `mld_starttimer()`, and `MLD_UNSOLICITED_REPORT_INTERVAL`.

Referenced by `in6_addmulti()`, and `mld_timeo()`.

Here is the call graph for this function:



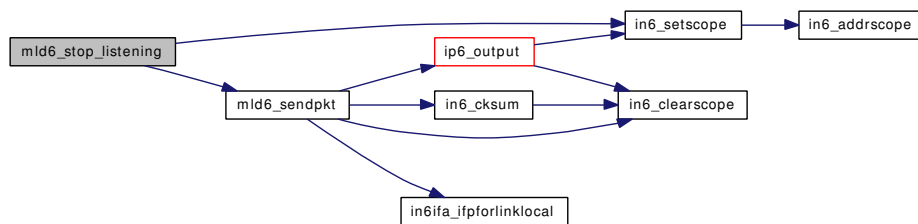
7.53.2.7 void mld6_stop_listening (struct **in6_multi** * *in6m*)

Definition at line 248 of file mld6.c.

References IN6_ARE_ADDR_EQUAL, in6_setscope(), in6addr_linklocal_allnodes, in6addr_linklocal_allrouters, IPV6_ADDR_MC_SCOPE, IPV6_ADDR_SCOPE_INTFACELOCAL, mld6_sendpkt(), and MLD_IREPORTEDLAST.

Referenced by in6_delmulti().

Here is the call graph for this function:



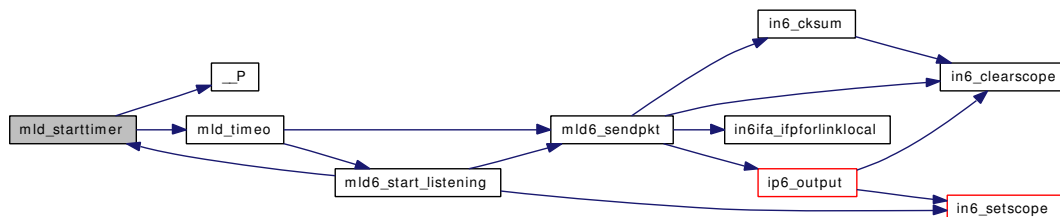
7.53.2.8 static void mld_starttimer (struct **in6_multi** *) [static]

Definition at line 134 of file mld6.c.

References __P(), and mld_timeo().

Referenced by in6_addmulti(), mld6_input(), and mld6_start_listening().

Here is the call graph for this function:



7.53.2.9 static void mld_stoptimer (struct **in6_multi** *) [static]

Definition at line 154 of file mld6.c.

References IN6M_TIMER_UNDEF.

Referenced by in6_delmulti(), and mld6_input().

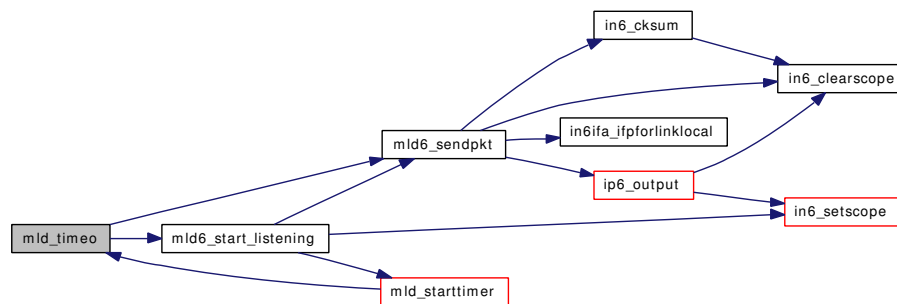
7.53.2.10 static void mld_timeo (struct in6_multi *) [static]

Definition at line 165 of file mld6.c.

References IN6M_TIMER_UNDEF, mld6_sendpkt(), mld6_start_listening(), and MLD_REPORTPENDING.

Referenced by mld_starttimer().

Here is the call graph for this function:



7.53.2.11 static u_long mld_timerresid (struct in6_multi *) [static]

Definition at line 187 of file mld6.c.

Referenced by mld6_input().

7.53.3 Variable Documentation

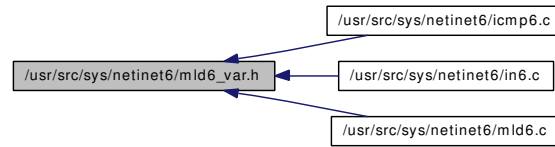
7.53.3.1 struct ip6_pktopts ip6_opts [static]

Definition at line 104 of file mld6.c.

Referenced by mld6_init(), and mld6_sendpkt().

7.54 /usr/src/sys/netinet6/mld6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define [MLD_RANDOM_DELAY\(X\)](#) (arc4random() % (X) + 1)
- #define [MLD_OTHERLISTENER](#) 0
- #define [MLD_IREPORTEDLAST](#) 1
- #define [MLD_REPORTPENDING](#) 2

Functions

- void [mld6_init](#) (void)
- void [mld6_input](#) (struct mbuf *, int)
- void [mld6_start_listening](#) (struct in6_multi *)
- void [mld6_stop_listening](#) (struct in6_multi *)
- void [mld6_fasttimeo](#) (void)

7.54.1 Define Documentation

7.54.1.1 #define MLD_IREPORTEDLAST 1

Definition at line 44 of file mld6_var.h.

Referenced by [mld6_input\(\)](#), [mld6_start_listening\(\)](#), and [mld6_stop_listening\(\)](#).

7.54.1.2 #define MLD_OTHERLISTENER 0

Definition at line 43 of file mld6_var.h.

Referenced by [mld6_input\(\)](#), and [mld6_start_listening\(\)](#).

7.54.1.3 #define MLD_RANDOM_DELAY(X) (arc4random() % (X) + 1)

Definition at line 38 of file mld6_var.h.

7.54.1.4 #define MLD_REPORTPENDING 2

Definition at line 45 of file mld6_var.h.

Referenced by [in6_addmulti\(\)](#), [in6_update_ifa\(\)](#), and [mld_timeo\(\)](#).

7.54.2 Function Documentation

7.54.2.1 void mld6_fasttimeo (void)

7.54.2.2 void mld6_init (void)

Definition at line 113 of file mld6.c.

References ip6_initpktopts(), ip6_opts, and ip6_pktopts::ip6po_hbh.

Referenced by icmp6_init().

Here is the call graph for this function:



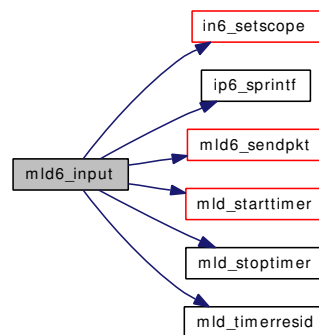
7.54.2.3 void mld6_input (struct mbuf *, int)

Definition at line 272 of file mld6.c.

References icmp6stat, IFP_TO_IA6, IN6_ARE_ADDR_EQUAL, IN6_IS_ADDR_LINKLOCAL, IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, IN6_LOOKUP_MULTI, in6_setscope(), in6addr_linklocal_allnodes, in6_multi::in6m_addr, in6_multi::in6m_state, in6_multi::in6m_timer, IN6M_TIMER_UNDEF, INET6_ADDRSTRLEN, ip6_sprintf(), IPV6_ADDR_MC_SCOPE, IPV6_ADDR_SCOPE_LINKLOCAL, M_LOOP, mld6_sendpkt(), MLD_IREPORTEDLAST, MLD_OTHERLISTENER, mld_starttimer(), mld_stoptimer(), MLD_TIMER_SCALE, and mld_timerresid().

Referenced by icmp6_input().

Here is the call graph for this function:



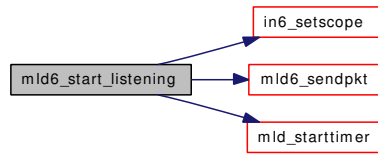
7.54.2.4 void mld6_start_listening (struct in6_multi *)

Definition at line 212 of file mld6.c.

References IN6_ARE_ADDR_EQUAL, in6_setscope(), in6addr_linklocal_allnodes, IPV6_ADDR_MC_SCOPE, IPV6_ADDR_SCOPE_LINKLOCAL, mld6_sendpkt(), MLD_IREPORTEDLAST, MLD_OTHERLISTENER, mld_starttimer(), and MLD_UNSOLICITED_REPORT_INTERVAL.

Referenced by in6_addmulti(), and mld_timeo().

Here is the call graph for this function:



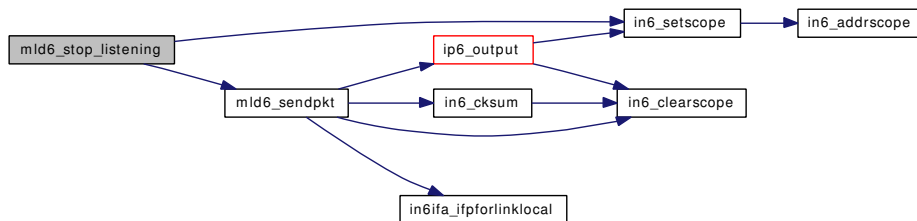
7.54.2.5 void mld6_stop_listening (struct in6_multi *)

Definition at line 248 of file mld6.c.

References `IN6_ARE_ADDR_EQUAL`, `in6_setscope()`, `in6addr_linklocal_allnodes`, `in6addr_linklocal_allrouters`, `IPV6_ADDR_MC_SCOPE`, `IPV6_ADDR_SCOPE_INTFACELOCAL`, `mld6_sendpkt()`, and `MLD_IREPORTEDLAST`.

Referenced by `in6_delmulti()`.

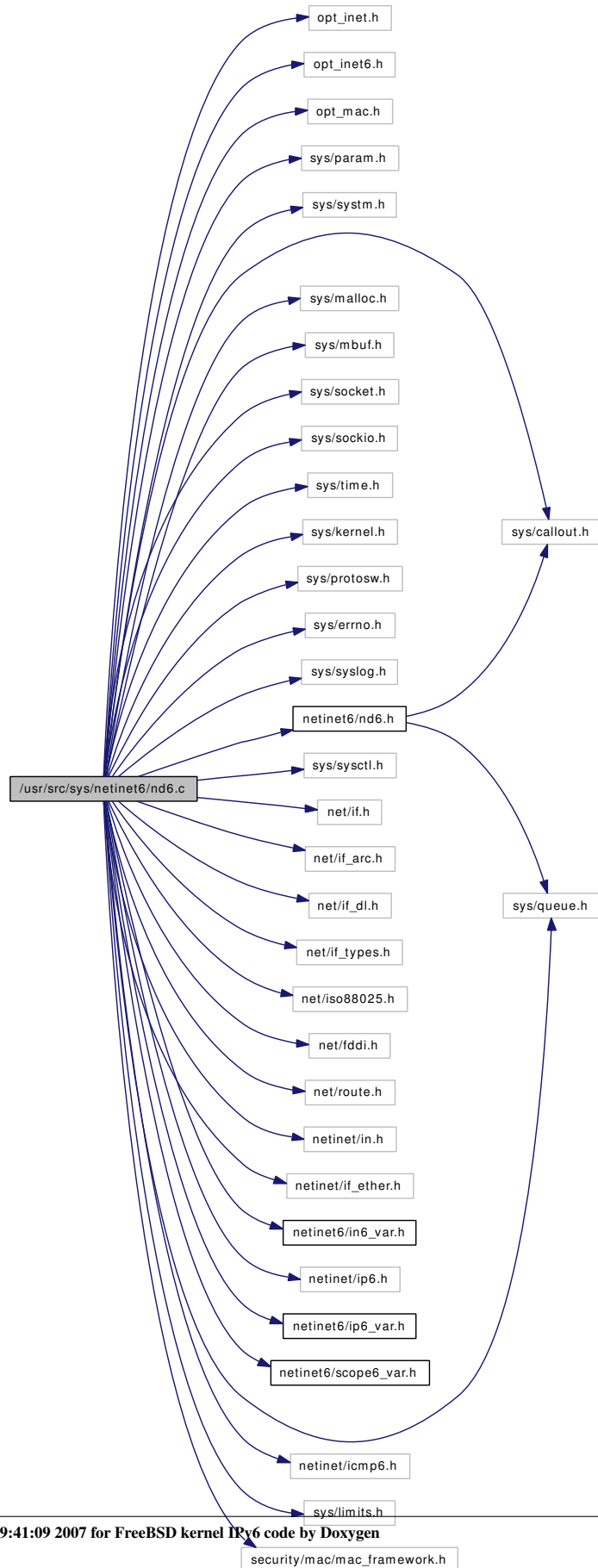
Here is the call graph for this function:



7.55 /usr/src/sys/netinet6/nd6.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_mac.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/callout.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/sockio.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/protosw.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/queue.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <net/if_arc.h>
#include <net/if_dl.h>
#include <net/if_types.h>
#include <net/iso88025.h>
#include <net/fddi.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/if_ether.h>
#include <netinet6/in6_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet6/nd6.h>
#include <netinet/icmp6.h>
#include <sys/limits.h>
#include <security/mac/mac_framework.h>
```

Include dependency graph for nd6.c:



Defines

- #define `ND6_SLOWTIMER_INTERVAL` (60 * 60)
- #define `ND6_RECALC_REACHTM_INTERVAL` (60 * 120)
- #define `SIN6(s)` ((struct `sockaddr_in6` *)s)
- #define `SDL(s)` ((struct `sockaddr_dl` *)s)
- #define `RTRADDR` `opr1` → `prefix[i].advrtr[j]`
- #define `ND` `ndi` → `ndi`
- #define `senderr(e)` { `error = (e); goto bad;`}

Functions

- static int `nd6_is_new_addr_neighbor` `__P` ((struct `sockaddr_in6` *, struct `ifnet` *)
- static void `nd6_setmtu0` `__P` ((struct `ifnet` *, struct `nd_ifinfo` *)
- static void `nd6_slowtimo` `__P` ((void *)
- static int `regen_tmpaddr` `__P` ((struct `in6_ifaddr` *)
- static struct `llinfo_nd6` *`nd6_free` `__P` ((struct `rtenry` *, int)
- static void `clear_llinfo_pqueue` `__P` ((struct `llinfo_nd6` *)
- void `nd6_init` ()
- `nd_ifinfo` * `nd6_ifattach` (struct `ifnet` *`ifp`)
- void `nd6_ifdetach` (struct `nd_ifinfo` *`nd`)
- void `nd6_setmtu` (struct `ifnet` *`ifp`)
- void `nd6_setmtu0` (struct `ifnet` *`ifp`, struct `nd_ifinfo` *`ndi`)
- void `nd6_option_init` (void *`opt`, int `icmplen`, union `nd_opts` *`ndopts`)
- `nd_opt_hdr` * `nd6_option` (union `nd_opts` *`ndopts`)
- int `nd6_options` (union `nd_opts` *`ndopts`)
- void `nd6_llinfo_settimer` (struct `llinfo_nd6` *`ln`, long `tick`)
- static void `nd6_llinfo_timer` (void *`arg`)
- void `nd6_timer` (void *`ignored_arg`)
- static int `regen_tmpaddr` (struct `in6_ifaddr` *`ia6`)
- void `nd6_purge` (struct `ifnet` *`ifp`)
- `rtenry` * `nd6_lookup` (struct `in6_addr` *`addr6`, int `create`, struct `ifnet` *`ifp`)
- static int `nd6_is_new_addr_neighbor` (struct `sockaddr_in6` *`addr`, struct `ifnet` *`ifp`)
- int `nd6_is_addr_neighbor` (struct `sockaddr_in6` *`addr`, struct `ifnet` *`ifp`)
- static struct `llinfo_nd6` * `nd6_free` (struct `rtenry` *`rt`, int `gc`)
- void `nd6_nud_hint` (struct `rtenry` *`rt`, struct `in6_addr` *`dst6`, int `force`)
- void `nd6_rrequest` (int `req`, struct `rtenry` *`rt`, struct `rt_addrinfo` *`info`)
- int `nd6_ioctl` (u_long `cmd`, `caddr_t` `data`, struct `ifnet` *`ifp`)
- `rtenry` * `nd6_cache_lladdr` (struct `ifnet` *`ifp`, struct `in6_addr` *`from`, char *`lladdr`, int `lladdrlen`, int `type`, int `code`)
- static void `nd6_slowtimo` (void *`ignored_arg`)
- int `nd6_output` (struct `ifnet` *`ifp`, struct `ifnet` *`origifp`, struct `mbuf` *`m0`, struct `sockaddr_in6` *`dst`, struct `rtenry` *`rt0`)
- int `nd6_need_cache` (struct `ifnet` *`ifp`)
- int `nd6_storelladdr` (struct `ifnet` *`ifp`, struct `rtenry` *`rt0`, struct `mbuf` *`m`, struct `sockaddr` *`dst`, u_char *`desten`)
- static void `clear_llinfo_pqueue` (struct `llinfo_nd6` *`ln`)
- static int `nd6_sysctl_drlist` (SYSCTL_HANDLER_ARGS)
- static int `nd6_sysctl_prlist` (SYSCTL_HANDLER_ARGS)

- **SYSCTL_NODE** (`_net_inet6_icmp6, ICMPV6CTL_ND6_DRLIST, nd6_drlist, CTLFLAG_RD, nd6_sysctl_drlist, ""`)
- **SYSCTL_NODE** (`_net_inet6_icmp6, ICMPV6CTL_ND6_PRLIST, nd6_prlist, CTLFLAG_RD, nd6_sysctl_prlist, ""`)
- **SYSCTL_INT** (`_net_inet6_icmp6, ICMPV6CTL_ND6_MAXQLEN, nd6_maxqueuelen, CTLFLAG_RW, &nd6_maxqueuelen, 1, ""`)

Variables

- `int nd6_prune = 1`
- `int nd6_delay = 5`
- `int nd6_umaxtries = 3`
- `int nd6_mmaxtries = 3`
- `int nd6_useloopback = 1`
- `int nd6_gctimer = (60 * 60 * 24)`
- `int nd6_maxndopt = 10`
- `int nd6_maxnudhint = 0`
- `int nd6_maxqueuelen = 1`
- `int nd6_debug = 0`
- `static int nd6_inuse`
- `static int nd6_allocated`
- `llinfo_nd6 llinfo_nd6 = { &llinfo_nd6, &llinfo_nd6 }`
- `nd_drhead nd_defrouter`
- `nd_prhead nd_prefix = { 0 }`
- `int nd6_recalc_reachtm_interval = ND6_RECALC_REACHTM_INTERVAL`
- `static struct sockaddr_in6 all1_sa`
- `callout nd6_slowtimo_ch`
- `callout nd6_timer_ch`
- `callout in6_tmpaddrtimer_ch`

7.55.1 Define Documentation

7.55.1.1 #define ND ndi → ndi

Referenced by `nd6_ioctl()`.

7.55.1.2 #define ND6_RECALC_REACHTM_INTERVAL (60 * 120)

Definition at line 74 of file `nd6.c`.

7.55.1.3 #define ND6_SLOWTIMER_INTERVAL (60 * 60)

Definition at line 73 of file `nd6.c`.

Referenced by `nd6_init()`, and `nd6_slowtimo()`.

7.55.1.4 #define RTRADDR oprl → prefix[i].advrtr[j]

Referenced by `nd6_ioctl()`.

7.55.1.5 #define SDL(s) ((struct sockaddr_dl *)s)

Definition at line 77 of file nd6.c.

Referenced by nd6_cache_lladdr(), nd6_na_input(), nd6_ns_input(), and nd6_storelladdr().

7.55.1.6 #define senderr(e) { error = (e); goto bad;}

Definition at line 1939 of file nd6.c.

Referenced by ip6_forward(), and nd6_output().

7.55.1.7 #define SIN6(s) ((struct sockaddr_in6 *)s)

Definition at line 76 of file nd6.c.

Referenced by nd6_storelladdr(), and rt6_deleteroute().

7.55.2 Function Documentation

7.55.2.1 static void clear_llinfo_pqueue __P ((struct llinfo_nd6 *)) [static]

7.55.2.2 static struct llinfo_nd6* nd6_free __P ((struct rentry *, int)) [static]

7.55.2.3 static int regen_tmpaddr __P ((struct in6_ifaddr *)) [static]

7.55.2.4 static void nd6_slowtimo __P ((void *)) [static]

7.55.2.5 static void nd6_setmtu0 __P ((struct ifnet *, struct nd_ifinfo *)) [static]

7.55.2.6 static int nd6_is_new_addr_neighbor __P ((struct sockaddr_in6 *, struct ifnet *)) [static]

7.55.2.7 static void clear_llinfo_pqueue (struct llinfo_nd6 *ln) [static]

Definition at line 2283 of file nd6.c.

Referenced by nd6_llinfo_timer().

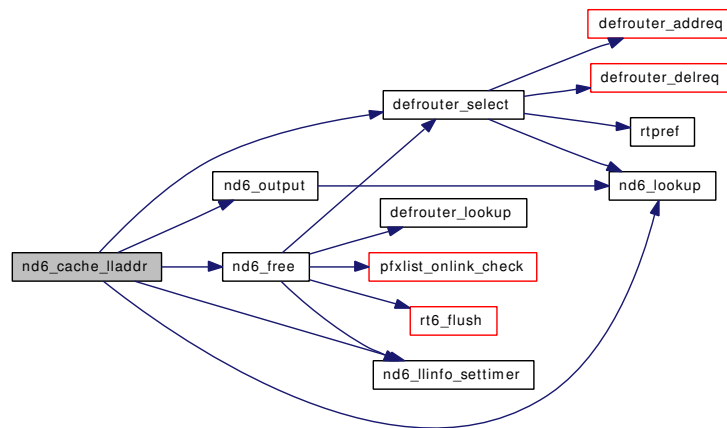
7.55.2.8 struct rentry* nd6_cache_lladdr (struct ifnet *ifp, struct in6_addr *from, char *lladdr, int lladdrlen, int type, int code)

Definition at line 1673 of file nd6.c.

References defrouter_select(), IN6_IS_ADDR_UNSPECIFIED, ip6_accept_rtadv, ip6_forwarding, llinfo_nd6::ln_router, llinfo_nd6::ln_state, nd6_free(), nd6_gctimer, ND6_LLINFO_INCOMPLETE, ND6_LLINFO_NOSTATE, nd6_llinfo_settimer(), ND6_LLINFO_STALE, nd6_lookup(), nd6_output(), and SDL.

Referenced by icmp6_redirect_input(), nd6_ns_input(), nd6_ra_input(), and nd6_rs_input().

Here is the call graph for this function:



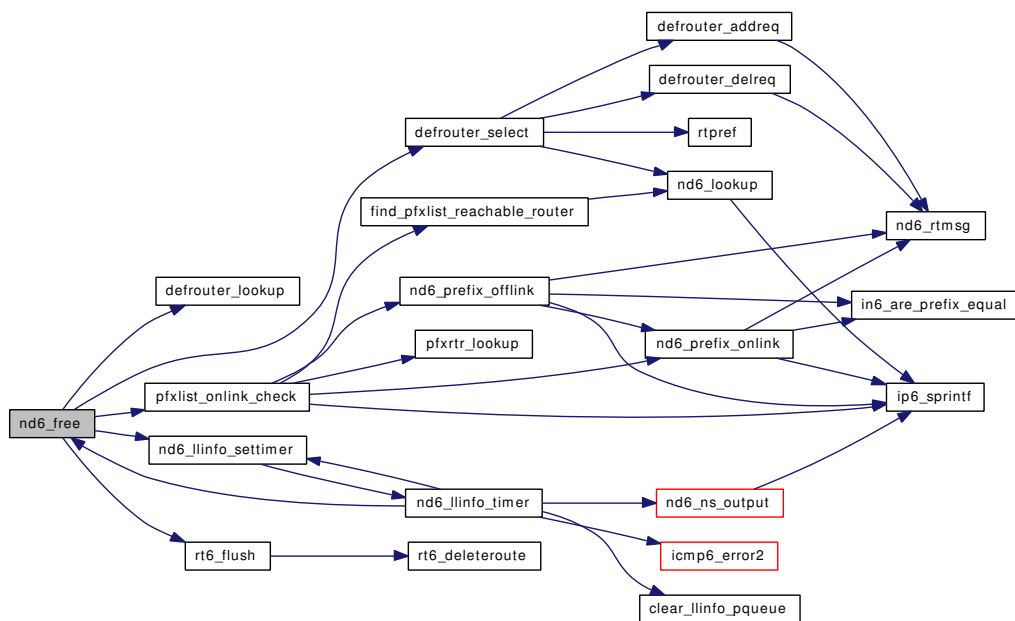
7.55.2.9 static struct `llinfo_nd6`* `nd6_free` (struct `rtentry` * `rt`, int `gc`) [static]

Definition at line 1029 of file `nd6.c`.

References `defrouter_lookup()`, `defrouter_select()`, `nd_defrouter::expire`, `ip6_forwarding`, `llinfo_nd6`, `llinfo_nd6::ln_next`, `llinfo_nd6::ln_router`, `llinfo_nd6::ln_state`, `nd6_gctimer`, `ND6_LLINFO_INCOMPLETE`, `nd6_llinfo_settimer()`, `ND6_LLINFO_STALE`, `pfplist_onlink_check()`, and `rt6_flush()`.

Referenced by `nd6_cache_lladdr()`, and `nd6_llinfo_timer()`.

Here is the call graph for this function:



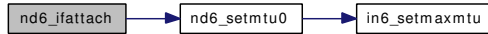
7.55.2.10 struct `nd_ifinfo*` `nd6_ifattach` (struct `ifnet * ifp`)

Definition at line 150 of file `nd6.c`.

References `ND6_IFF_ACCEPT_RTADV`, `ND6_IFF_PERFORMNUD`, `nd6_setmtu0()`, `ND_COMPUTE_RT`, `ND_COMPUTE_RT`, `REACHABLE_TIME`, and `RETRANS_TIMER`.

Referenced by `in6_domifattach()`.

Here is the call graph for this function:



7.55.2.11 void `nd6_ifdetach` (struct `nd_ifinfo * nd`)

Definition at line 178 of file `nd6.c`.

Referenced by `in6_domifdetach()`.

7.55.2.12 void `nd6_init` ()

Definition at line 123 of file `nd6.c`.

References `all1_sa`, `ND6_SLOWTIMER_INTERVAL`, `nd6_slowtimo()`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, and `sockaddr_in6::sin6_len`.

Here is the call graph for this function:



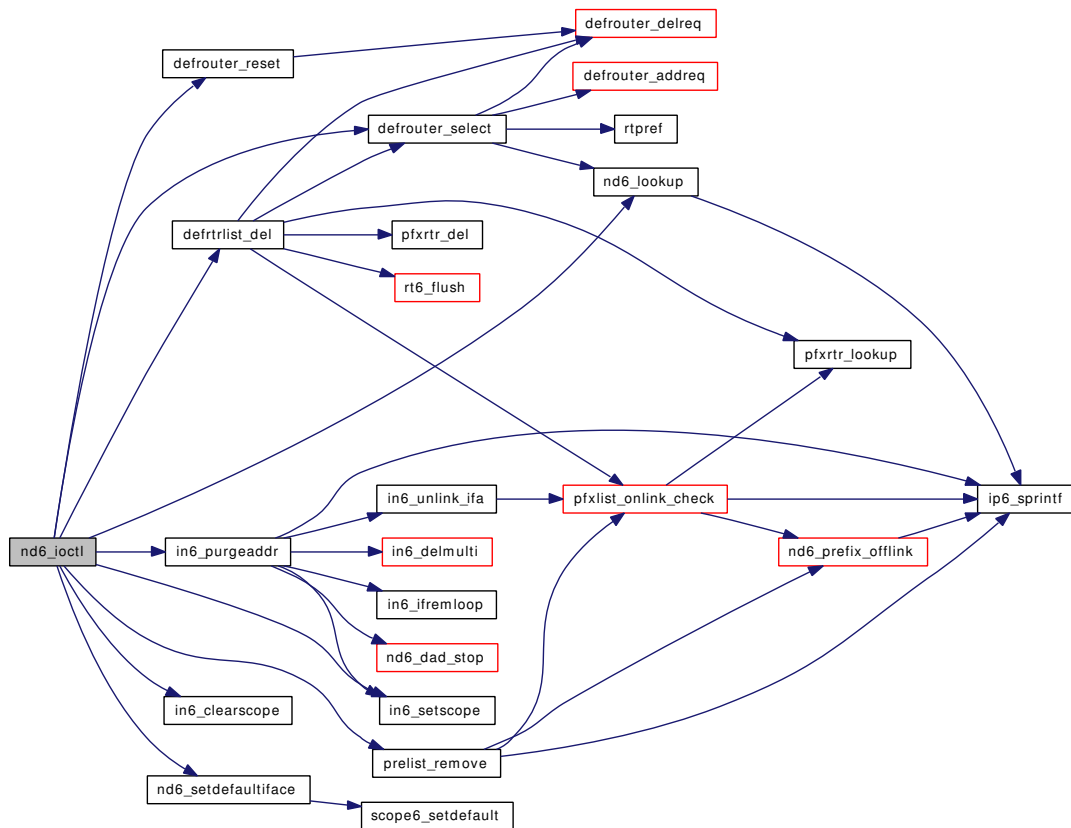
7.55.2.13 int `nd6_ioctl` (u_long `cmd`, caddr_t `data`, struct `ifnet * ifp`)

Definition at line 1436 of file `nd6.c`.

References `in6_nbrinfo::addr`, `in6_nbrinfo::asked`, `defrouter_reset()`, `defrouter_select()`, `defrrlist_del()`, `DRLSTSIZ`, `in6_nbrinfo::expire`, `nd_defrouter::expire`, `nd_defrouter::flags`, `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia6_ndpr`, `in6_ifaddr::ia_ifa`, `in6_ifaddr::ia_next`, `in6_ndifreq::ifindex`, `nd_defrouter::ifp`, `in6_clearscope()`, `IN6_IFF_AUTOCONF`, `IN6_IS_ADDR_LINKLOCAL`, `IN6_LINKMTU`, `in6_purgeaddr()`, `in6_setscope()`, `in6_nbrinfo::isrouter`, `llinfo_nd6::ln_asked`, `llinfo_nd6::ln_expire`, `llinfo_nd6::ln_router`, `llinfo_nd6::ln_state`, `ND`, `nd_defifindex`, `ND6_INFINITE_LIFETIME`, `nd6_lookup()`, `nd6_setdefaultiface()`, `ND_COMPUTE_RT`, `ND_COMPUTE_RT`, `ND_IFINFO`, `in6_ndireq::ndi`, `nd_prefix::ndpr_ifp`, `nd_prefix::ndpr_lastupdate`, `nd_prefix::ndpr_plen`, `nd_prefix::ndpr_pltime`, `nd_prefix::ndpr_prefix`, `nd_prefix::ndpr_vltime`, `OSIOCGIFINFO_IN6`, `PR_ORIG_RA`, `prelist_remove()`, `PRLSTSIZ`, `nd_pfxrouter::router`, `nd_defrouter::rtaddr`, `nd_defrouter::rtlifetime`, `RTRADDR`, `sockaddr_in6::sin6_addr`, `SIOCGDEFIFACE_IN6`, `SIOCGDRLST_IN6`, `SIOCGIFINFO_IN6`, `SIOCGNBRINFO_IN6`, `SIOCGPRLST_IN6`, `SIOCSDEFIFACE_IN6`, `SIOCSIFINFO_FLAGS`, `SIOCSIFINFO_IN6`, `SIOCSNDFLUSH_IN6`, `SIOCSPFXFLUSH_IN6`, `SIOCSRTRFLUSH_IN6`, and `in6_nbrinfo::state`.

Referenced by `in6_control()`.

Here is the call graph for this function:



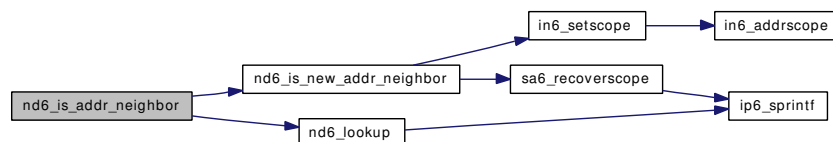
7.55.2.14 int nd6_is_addr_neighbor (struct sockaddr_in6 * addr, struct ifnet * ifp)

Definition at line 1004 of file nd6.c.

References `nd6_is_new_addr_neighbor()`, and `nd6_lookup()`.

Referenced by `icmp6_redirect_output()`, and `nd6_output()`.

Here is the call graph for this function:



7.55.2.15 static int nd6_is_new_addr_neighbor (struct sockaddr_in6 * addr, struct ifnet * ifp) [static]

Definition at line 928 of file nd6.c.

References `IN6_ARE_MASKED_ADDR_EQUAL`, `IN6_IS_ADDR_LINKLOCAL`, `in6_setscope()`, `ip6_forwarding`, `nd6_defifindex`, `nd_prefix::ndpr_ifp`, `nd_prefix::ndpr_mask`, `nd_prefix::ndpr_prefix`, `nd_`

prefix::ndpr_stateflags, NDPRF_ONLINK, sa6_recoverscope(), sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_scope_id.

Referenced by nd6_is_addr_neighbor().

Here is the call graph for this function:



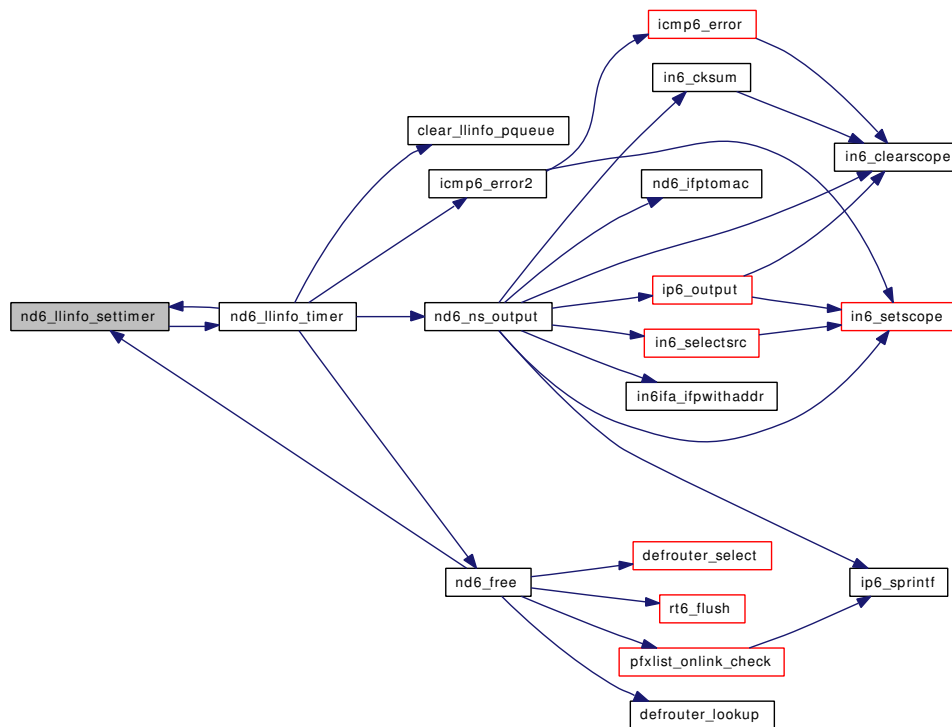
7.55.2.16 void nd6_llinfo_settimer (struct llinfo_nd6 * ln, long tick)

Definition at line 394 of file nd6.c.

References llinfo_nd6::ln_expire, and nd6_llinfo_timer().

Referenced by nd6_cache_lladdr(), nd6_free(), nd6_llinfo_timer(), nd6_na_input(), and nd6_nud_hint().

Here is the call graph for this function:



7.55.2.17 static void nd6_llinfo_timer (void * arg) [static]

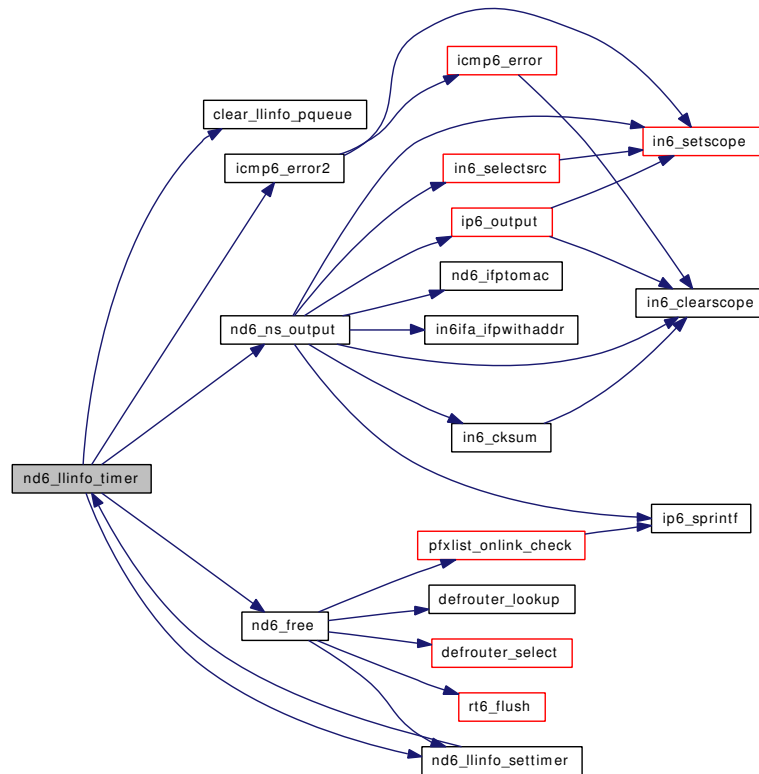
Definition at line 417 of file nd6.c.

References clear_llinfo_pqueue(), nd_ifinfo::flags, icmp6_error2(), llinfo_nd6, llinfo_nd6::ln_asking, llinfo_nd6::ln_expire, llinfo_nd6::ln_hold, llinfo_nd6::ln_ntick, llinfo_nd6::ln_rt, llinfo_

nd6::ln_state, nd6_free(), nd6_gctimer, ND6_IFF_PERFORMNUD, ND6_LINFO_DELAY, ND6_LINFO_INCOMPLETE, ND6_LINFO_PERMANENT, ND6_LINFO_PROBE, ND6_LINFO_REACHABLE, nd6_llinfo_settimer(), ND6_LINFO_STALE, nd6_mmaxtries, nd6_ns_output(), nd6_umaxtries, ND_IFINFO, nd_ifinfo::retrans, and sockaddr_in6::sin6_addr.

Referenced by nd6_llinfo_settimer().

Here is the call graph for this function:



7.55.2.18 struct rentry* nd6_lookup (struct in6_addr * addr6, int create, struct ifnet * ifp)

Definition at line 820 of file nd6.c.

References all1_sa, INET6_ADDRSTRLEN, ip6_sprintf(), linfo_nd6, linfo_nd6::ln_state, ND6_LINFO_NOSTATE, nd6log, and sin6.

Referenced by defrouter_select(), find_pfxlist_reachable_router(), icmp6_redirect_output(), nd6_cache_lladdr(), nd6_ioctl(), nd6_is_addr_neighbor(), nd6_na_input(), nd6_nud_hint(), and nd6_output().

Here is the call graph for this function:



7.55.2.19 int nd6_need_cache (struct ifnet * ifp)

Definition at line 2171 of file nd6.c.

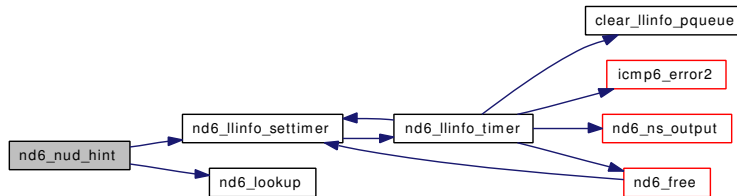
Referenced by in6_ifinit(), nd6_output(), and nd6_prefix_onlink().

7.55.2.20 void nd6_nud_hint (struct rentry * rt, struct in6_addr * dst6, int force)

Definition at line 1140 of file nd6.c.

References llinfo_nd6, llinfo_nd6::ln_byhint, llinfo_nd6::ln_state, ND6_LLINFO_PERMANENT, ND6_LLINFO_REACHABLE, nd6_llinfo_settimer(), nd6_lookup(), nd6_maxnudhint, and ND_IFINFO.

Here is the call graph for this function:



7.55.2.21 struct nd_opt_hdr* nd6_option (union nd_opts * ndopts)

Definition at line 262 of file nd6.c.

Referenced by nd6_options().

7.55.2.22 void nd6_option_init (void * opt, int icmp6len, union nd_opts * ndopts)

Definition at line 241 of file nd6.c.

Referenced by icmp6_redirect_input(), nd6_na_input(), nd6_ns_input(), nd6_ra_input(), and nd6_rs_input().

7.55.2.23 int nd6_options (union nd_opts * ndopts)

Definition at line 314 of file nd6.c.

References icmp6stat, nd6_maxndopt, nd6_option(), and nd6log.

Referenced by icmp6_redirect_input(), nd6_na_input(), nd6_ns_input(), nd6_ra_input(), and nd6_rs_input().

Here is the call graph for this function:



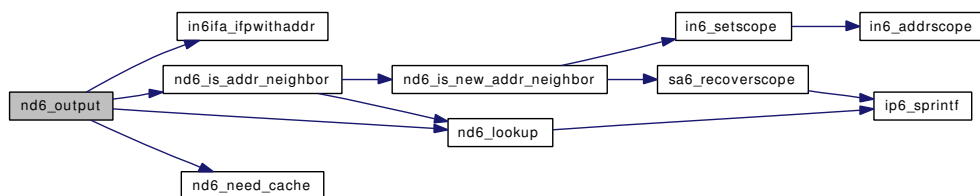
7.55.2.24 int nd6_output (struct ifnet * ifp, struct ifnet * origifp, struct mbuf * m0, struct sockaddr_in6 * dst, struct rentry * rt0)

Definition at line 1941 of file nd6.c.

References IN6_IS_ADDR_MULTICAST, in6ifa_ifpwithaddr(), nd6_is_addr_neighbor(), nd6_lookup(), nd6_need_cache(), senderr, and sockaddr_in6::sin6_addr.

Referenced by ip6_forward(), ip6_output(), nd6_cache_lladdr(), and nd6_na_input().

Here is the call graph for this function:



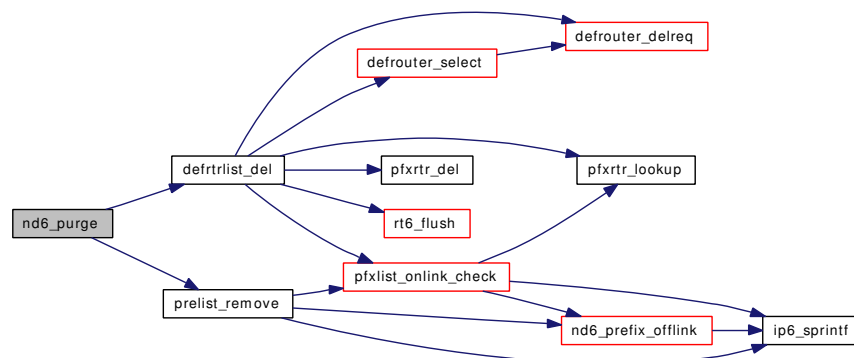
7.55.2.25 void nd6_purge (struct ifnet * ifp)

Definition at line 732 of file nd6.c.

References defrtrlist_del(), nd_defrouter::ifp, nd_defrouter::installed, llinfo_nd6, nd_prefix::ndpr_ifp, nd_prefix::ndpr_refcnt, and prelist_remove().

Referenced by in6_ifdetach().

Here is the call graph for this function:



7.55.2.26 void nd6_rtrequest (int req, struct rentry * rt, struct rt_addrinfo * info)

Definition at line 1188 of file nd6.c.

References llinfo_nd6.

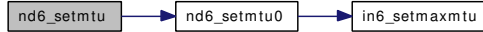
Referenced by in6_ifinit().

7.55.2.27 void nd6_setmtu (struct ifnet * ifp)

Definition at line 190 of file nd6.c.

References nd6_setmtu0(), and ND_IFINFO.

Here is the call graph for this function:



7.55.2.28 void nd6_setmtu0 (struct ifnet * ifp, struct nd_ifinfo * ndi)

Definition at line 199 of file nd6.c.

References in6_maxmtu, in6_setmaxmtu(), and nd_ifinfo::maxmtu.

Referenced by nd6_ifattach(), and nd6_setmtu().

Here is the call graph for this function:



7.55.2.29 static void nd6_slowtimo (void * ignored_arg) [static]

Definition at line 1913 of file nd6.c.

References nd_ifinfo::basereachable, nd6_recalc_reachtm_interval, ND6_SLOWTIMER_INTERVAL, ND_COMPUTE_RTTIME, ND_IFINFO, nd_ifinfo::reachable, and nd_ifinfo::recalctm.

Referenced by nd6_init().

7.55.2.30 int nd6_storelladdr (struct ifnet * ifp, struct rtenry * rt0, struct mbuf * m, struct sockaddr * dst, u_char * desten)

Definition at line 2207 of file nd6.c.

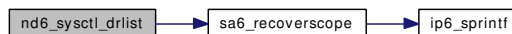
References SDL, and SIN6.

7.55.2.31 static int nd6_sysctl_drlist (SYSCTL_HANDLER_ARGS) [static]

Definition at line 2311 of file nd6.c.

References nd_defrouter::expire, nd_defrouter::flags, nd_defrouter::ifp, nd_defrouter::rtaddr, nd_defrouter::rtlifetime, and sa6_recoverscope().

Here is the call graph for this function:

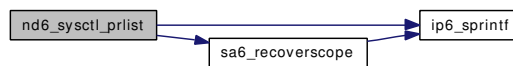


7.55.2.32 static int nd6_sysctl_prlist (SYSCTL_HANDLER_ARGS) [static]

Definition at line 2349 of file nd6.c.

References INET6_ADDRSTRLEN, ip6_sprintf(), ND6_INFINITE_LIFETIME, nd_prefix::ndpr_ifp, nd_prefix::ndpr_lastupdate, nd_prefix::ndpr_plen, nd_prefix::ndpr_pltime, nd_prefix::ndpr_prefix, nd_prefix::ndpr_refcnt, nd_prefix::ndpr_stateflags, nd_prefix::ndpr_vltime, PR_ORIG_RA, nd_pfxrouter::router, nd_defrouter::rtaddr, sa6_recoverscope(), and sin6.

Here is the call graph for this function:



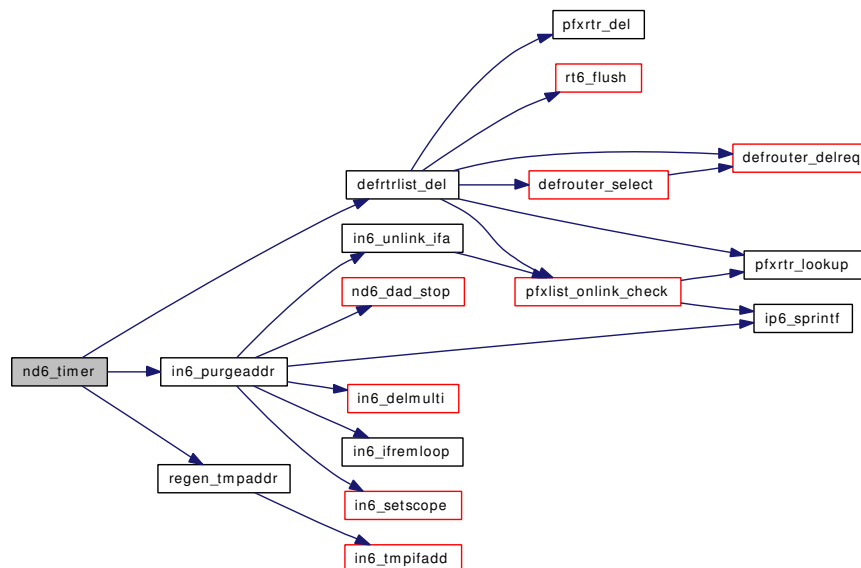
7.55.2.33 void nd6_timer (void * ignored_arg)

Definition at line 540 of file nd6.c.

References defrtrlist_del(), nd_defrouter::expire, in6_ifaddr::ia6_flags, in6_ifaddr::ia6_lifetime, in6_ifaddr::ia_ifa, in6_ifaddr::ia_next, IFA6_IS_DEPRECATED, IFA6_IS_INVALID, IN6_IFF_DEPRECATED, IN6_IFF_TEMPORARY, in6_purgeaddr(), ip6_use_tempaddr, nd6_prune, and regen_tmpaddr().

Referenced by ip6_init2().

Here is the call graph for this function:



7.55.2.34 static int regen_tmpaddr (struct in6_ifaddr * ia6) [static]

Definition at line 665 of file nd6.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia6_ndpr`, `IFA6_IS_DEPRECATED`, `IN6_IFF_AUTOCONF`, `IN6_IFF_TEMPORARY`, and `in6_tmpifadd()`.

Referenced by `nd6_timer()`.

Here is the call graph for this function:



7.55.2.35 `SYSCTL_INT` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_MAXQLEN`, `nd6_maxqueuelen`, `CTLFLAG_RW`, `&nd6_maxqueuelen`, `1`, `""`)

7.55.2.36 `SYSCTL_NODE` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_PRLIST`, `nd6_prlist`, `CTLFLAG_RD`, `nd6_sysctl_prlist`, `""`)

7.55.2.37 `SYSCTL_NODE` (`_net_inet6_icmp6`, `ICMPV6CTL_ND6_DRLIST`, `nd6_drlist`, `CTLFLAG_RD`, `nd6_sysctl_drlist`, `""`)

7.55.3 Variable Documentation

7.55.3.1 `struct sockaddr_in6 all1_sa` [`static`]

Definition at line 107 of file `nd6.c`.

Referenced by `in6_ifloop_request()`, `nd6_init()`, and `nd6_lookup()`.

7.55.3.2 `struct callout in6_tmpaddrtimer_ch`

Definition at line 69 of file `in6_ifattach.c`.

Referenced by `in6_tmpaddrtimer()`, and `ip6_init2()`.

7.55.3.3 `struct llinfo_nd6 llinfo_nd6 = {&llinfo_nd6, &llinfo_nd6}`

Definition at line 102 of file `nd6.c`.

Referenced by `nd6_free()`, `nd6_llinfo_timer()`, `nd6_lookup()`, `nd6_nud_hint()`, `nd6_purge()`, and `nd6_rrequest()`.

7.55.3.4 `int nd6_allocated` [`static`]

Definition at line 100 of file `nd6.c`.

7.55.3.5 `int nd6_debug = 0`

Definition at line 96 of file `nd6.c`.

7.55.3.6 `int nd6_delay = 5`

Definition at line 81 of file `nd6.c`.

7.55.3.7 int nd6_gctimer = (60 * 60 * 24)

Definition at line 85 of file nd6.c.

Referenced by nd6_cache_lladdr(), nd6_free(), nd6_llinfo_timer(), and nd6_na_input().

7.55.3.8 int nd6_inuse [static]

Definition at line 100 of file nd6.c.

7.55.3.9 int nd6_maxndopt = 10

Definition at line 88 of file nd6.c.

Referenced by nd6_options().

7.55.3.10 int nd6_maxnudhint = 0

Definition at line 90 of file nd6.c.

Referenced by nd6_nud_hint().

7.55.3.11 int nd6_maxqueuelen = 1

Definition at line 91 of file nd6.c.

7.55.3.12 int nd6_mmaxtries = 3

Definition at line 83 of file nd6.c.

Referenced by nd6_llinfo_timer().

7.55.3.13 int nd6_prune = 1

Definition at line 80 of file nd6.c.

Referenced by nd6_timer().

7.55.3.14 int nd6_recalc_reachtm_interval = ND6_RECALC_REACHTM_INTERVAL

Definition at line 106 of file nd6.c.

Referenced by nd6_slowtimo().

7.55.3.15 struct callout nd6_slowtimo_ch

Definition at line 118 of file nd6.c.

7.55.3.16 struct callout [nd6_timer_ch](#)

Definition at line 119 of file nd6.c.

Referenced by ip6_init2().

7.55.3.17 int [nd6_umaxtries](#) = 3

Definition at line 82 of file nd6.c.

Referenced by nd6_llinfo_timer().

7.55.3.18 int [nd6_uselookback](#) = 1

Definition at line 84 of file nd6.c.

7.55.3.19 struct nd_drhead [nd_defrouter](#)

Definition at line 103 of file nd6.c.

7.55.3.20 struct nd_prhead [nd_prefix](#) = { 0 }

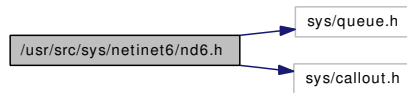
Definition at line 104 of file nd6.c.

7.56 /usr/src/sys/netinet6/nd6.h File Reference

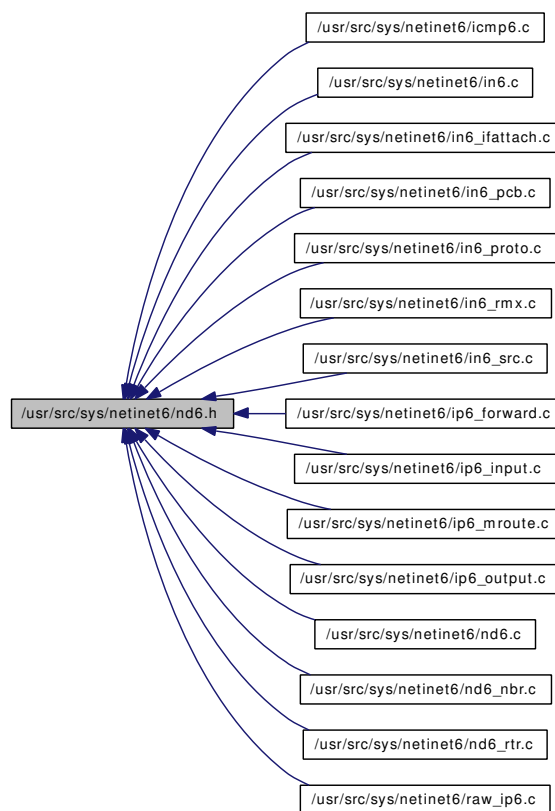
```
#include <sys/queue.h>
```

```
#include <sys/callout.h>
```

Include dependency graph for nd6.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [llinfo_nd6](#)
- struct [nd_ifinfo](#)
- struct [in6_nbrinfo](#)
- struct [in6_drlist](#)
- struct [in6_defrouter](#)
- struct [in6_oprlist](#)
- struct [in6_prlist](#)
- struct [in6_prefix](#)

- struct `in6_ondireq`
- struct `in6_ndireq`
- struct `in6_ndifreq`
- struct `nd_defrouter`
- struct `nd_prefixctl`
- struct `nd_prefix`
- struct `inet6_ndpr_msghdr`
- struct `nd_pfxrouter`
- union `nd_opts`

Defines

- #define `RTF_ANNOUNCE` `RTF_PROTO2`
- #define `ND6_LLINFO_NOSTATE` `-2`
- #define `ND6_LLINFO_INCOMPLETE` `0`
- #define `ND6_LLINFO_REACHABLE` `1`
- #define `ND6_LLINFO_STALE` `2`
- #define `ND6_LLINFO_DELAY` `3`
- #define `ND6_LLINFO_PROBE` `4`
- #define `ND6_IS_LLINFO_PROBREACH`(n) `((n) → ln_state > ND6_LLINFO_INCOMPLETE)`
- #define `ND6_LLINFO_PERMANENT`(n) `((n) → ln_expire == 0) && ((n) → ln_state > ND6_LLINFO_INCOMPLETE)`
- #define `ND6_IFF_PERFORMNUD` `0x1`
- #define `ND6_IFF_ACCEPT_RTADV` `0x2`
- #define `ND6_IFF_PREFER_SOURCE` `0x4`
- #define `ND6_IFF_IFDISABLED` `0x8`
- #define `ND6_IFF_DONT_SET_IFROUTE` `0x10`
- #define `ND_IFINFO`(ifp) `((struct in6_ifextra *) (ifp) → if_afdata[AF_INET6]) → nd_ifinfo`
- #define `IN6_LINKMTU`(ifp)
- #define `DRLSTSIZ` `10`
- #define `PRLSTSIZ` `10`
- #define `NDPRF_ONLINK` `0x1`
- #define `NDPRF_DETACHED` `0x2`
- #define `MAX_RTR_SOLICITATION_DELAY` `1`
- #define `RTR_SOLICITATION_INTERVAL` `4`
- #define `MAX_RTR_SOLICITATIONS` `3`
- #define `ND6_INFINITE_LIFETIME` `0xffffffff`
- #define `MAX_REACHABLE_TIME` `3600000`
- #define `REACHABLE_TIME` `30000`
- #define `RETRANS_TIMER` `1000`
- #define `MIN_RANDOM_FACTOR` `512`
- #define `MAX_RANDOM_FACTOR` `1536`
- #define `DEF_TEMP_VALID_LIFETIME` `604800`
- #define `DEF_TEMP_PREFERRED_LIFETIME` `86400`
- #define `TEMPADDR_REGEN_ADVANCE` `5`
- #define `MAX_TEMP_DESYNC_FACTOR` `600`
- #define `ND_COMPUTE_RUNTIME`(x)
- #define `ndpr_next` `ndpr_entry.le_next`
- #define `ndpr_raf` `ndpr_flags`
- #define `ndpr_raf_onlink` `ndpr_flags.onlink`

- #define `ndpr_raf_auto` `ndpr_flags.autonomous`
- #define `ndpr_raf_router` `ndpr_flags.router`
- #define `prm_raf_onlink` `prm_flags.prf_ra.onlink`
- #define `prm_raf_auto` `prm_flags.prf_ra.autonomous`
- #define `prm_statef_onlink` `prm_flags.prf_state.onlink`
- #define `prm_rrf_decrvalid` `prm_flags.prf_rr.decrvalid`
- #define `prm_rrf_decrprefd` `prm_flags.prf_rr.decrprefd`
- #define `pfr_next` `pfr_entry.le_next`
- #define `nd6log(x)` `do { if (nd6_debug) log x; } while (/*CONSTCOND*/ 0)`
- #define `nd_opts_src_lladdr` `nd_opt_each.src_lladdr`
- #define `nd_opts_tgt_lladdr` `nd_opt_each.tgt_lladdr`
- #define `nd_opts_pi` `nd_opt_each.pi_beg`
- #define `nd_opts_pi_end` `nd_opt_each.pi_end`
- #define `nd_opts_rh` `nd_opt_each.rh`
- #define `nd_opts_mtu` `nd_opt_each.mtu`
- #define `nd_opts_search` `nd_opt_each.search`
- #define `nd_opts_last` `nd_opt_each.last`
- #define `nd_opts_done` `nd_opt_each.done`

Functions

- `TAILQ_HEAD` (`nd_drhead`, `nd_defrouter`)
- `LIST_HEAD` (`nd_prhead`, `nd_prefix`)
- `void nd6_init __P ((void))`
- `nd_ifinfo *nd6_ifattach __P ((struct ifnet *))`
- `void nd6_ifdetach __P ((struct nd_ifinfo *))`
- `int nd6_is_addr_neighbor __P ((struct sockaddr_in6 *, struct ifnet *))`
- `void nd6_option_init __P ((void *, int, union nd_opts *))`
- `nd_opt_hdr *nd6_option __P ((union nd_opts *))`
- `rtentry *nd6_lookup __P ((struct in6_addr *, int, struct ifnet *))`
- `void nd6_llinfo_settimer __P ((struct llinfo_nd6 *, long))`
- `void nd6_timer __P ((void *))`
- `void nd6_nud_hint __P ((struct rtentry *, struct in6_addr *, int))`
- `int nd6_resolve __P ((struct ifnet *, struct rtentry *, struct mbuf *, struct sockaddr *, u_char *))`
- `void nd6_rtrrequest __P ((int, struct rtentry *, struct rt_addrinfo *))`
- `int nd6_ioctl __P ((u_long, caddr_t, struct ifnet *))`
- `rtentry *nd6_cache_lladdr __P ((struct ifnet *, struct in6_addr *, char *, int, int, int))`
- `int nd6_output __P ((struct ifnet *, struct ifnet *, struct mbuf *, struct sockaddr_in6 *, struct rtentry *))`
- `void nd6_na_input __P ((struct mbuf *, int, int))`
- `void nd6_na_output __P ((struct ifnet *, const struct in6_addr *, const struct in6_addr *, u_long, int, struct sockaddr *))`
- `void nd6_ns_output __P ((struct ifnet *, const struct in6_addr *, const struct in6_addr *, struct llinfo_nd6 *, int))`
- `void nd6_dad_start __P ((struct ifaddr *, int))`
- `void nd6_dad_stop __P ((struct ifaddr *))`
- `void prelist_del __P ((struct nd_prefix *))`
- `void defrouter_addreq __P ((struct nd_defrouter *))`
- `int nd6_prelist_add __P ((struct nd_prefixctl *, struct nd_defrouter *, struct nd_prefix **))`
- `nd_defrouter *defrouter_lookup __P ((struct in6_addr *, struct ifnet *))`
- `nd_prefix *nd6_prefix_lookup __P ((struct nd_prefixctl *))`
- `int nd6_setdefaultiface __P ((int))`
- `int in6_tmpifadd __P ((const struct in6_ifaddr *, int, int))`

Variables

- int [nd6_prune](#)
- int [nd6_delay](#)
- int [nd6_umaxtries](#)
- int [nd6_mmaxtries](#)
- int [nd6_uselookback](#)
- int [nd6_maxnudhint](#)
- int [nd6_gctimer](#)
- [llinfo_nd6](#) [llinfo_nd6](#)
- [nd_drhead](#) [nd_defrouter](#)
- [nd_prhead](#) [nd_prefix](#)
- int [nd6_debug](#)
- callout [nd6_timer_ch](#)
- int [nd6_defifindex](#)
- int [ip6_desync_factor](#)
- [u_int32_t](#) [ip6_temp_preferred_lifetime](#)
- [u_int32_t](#) [ip6_temp_valid_lifetime](#)
- int [ip6_temp_regen_advance](#)

7.56.1 Define Documentation

7.56.1.1 #define DEF_TEMP_PREFERRED_LIFETIME 86400

Definition at line 236 of file nd6.h.

7.56.1.2 #define DEF_TEMP_VALID_LIFETIME 604800

Definition at line 235 of file nd6.h.

7.56.1.3 #define DRLSTSIZ 10

Definition at line 120 of file nd6.h.

Referenced by [nd6_ioctl\(\)](#).

7.56.1.4 #define IN6_LINKMTU(ifp)

Value:

```
((ND_IFINFO(ifp)->linkmtu && ND_IFINFO(ifp)->linkmtu < (ifp)->if_mtu) \
 ? ND_IFINFO(ifp)->linkmtu \
 : ((ND_IFINFO(ifp)->maxmtu && ND_IFINFO(ifp)->maxmtu < (ifp)->if_mtu) \
 ? ND_IFINFO(ifp)->maxmtu : (ifp)->if_mtu)
```

Definition at line 104 of file nd6.h.

Referenced by [in6_addroute\(\)](#), [in6_setmaxmtu\(\)](#), [ip6_forward\(\)](#), [ip6_getpmtu\(\)](#), [ip6_output\(\)](#), [nd6_ioctl\(\)](#), and [phyint_send\(\)](#).

7.56.1.5 #define MAX_RANDOM_FACTOR 1536

Definition at line 234 of file nd6.h.

7.56.1.6 #define MAX_REACHABLE_TIME 3600000

Definition at line 230 of file nd6.h.

Referenced by nd6_ra_input().

7.56.1.7 #define MAX_RTR_SOLICITATION_DELAY 1

Definition at line 222 of file nd6.h.

Referenced by in6_if_up(), and in6_update_ifa().

7.56.1.8 #define MAX_RTR_SOLICITATIONS 3

Definition at line 224 of file nd6.h.

7.56.1.9 #define MAX_TEMP_DESYNC_FACTOR 600

Definition at line 238 of file nd6.h.

7.56.1.10 #define MIN_RANDOM_FACTOR 512

Definition at line 233 of file nd6.h.

7.56.1.11 #define ND6_IFF_ACCEPT_RTADV 0x2

Definition at line 94 of file nd6.h.

Referenced by nd6_ifattach(), and nd6_ra_input().

7.56.1.12 #define ND6_IFF_DONT_SET_IFROUTE 0x10

Definition at line 99 of file nd6.h.

7.56.1.13 #define ND6_IFF_IFDISABLED 0x8

Definition at line 96 of file nd6.h.

Referenced by ip6_input(), and nd6_dad_duplicated().

7.56.1.14 #define ND6_IFF_PERFORMNUD 0x1

Definition at line 93 of file nd6.h.

Referenced by nd6_ifattach(), and nd6_llinfo_timer().

7.56.1.15 #define ND6_IFF_PREFER_SOURCE 0x4

Definition at line 95 of file nd6.h.

7.56.1.16 #define ND6_INFINITE_LIFETIME 0xffffffff

Definition at line 226 of file nd6.h.

Referenced by `in6_control()`, `in6_ifattach_linklocal()`, `in6_ifattach_loopback()`, `in6_init_address_ltimes()`, `in6_init_prefix_ltimes()`, `in6_update_ifa()`, `nd6_ioctl()`, `nd6_sysctl_prlist()`, `ni6_store_addrs()`, and `prelist_update()`.

7.56.1.17 #define ND6_IS_LLININFO_PROBREACH(n) ((n) → ln_state > ND6_LLININFO_INCOMPLETE)

Definition at line 74 of file nd6.h.

Referenced by `defrouter_select()`, and `find_pfxlist_reachable_router()`.

7.56.1.18 #define ND6_LLININFO_DELAY 3

Definition at line 71 of file nd6.h.

Referenced by `nd6_llinfo_timer()`.

7.56.1.19 #define ND6_LLININFO_INCOMPLETE 0

Definition at line 68 of file nd6.h.

Referenced by `nd6_cache_lladdr()`, `nd6_free()`, `nd6_llinfo_timer()`, and `nd6_na_input()`.

7.56.1.20 #define ND6_LLININFO_NOSTATE -2

Definition at line 59 of file nd6.h.

Referenced by `nd6_cache_lladdr()`, and `nd6_lookup()`.

7.56.1.21 #define ND6_LLININFO_PERMANENT(n) (((n) → ln_expire == 0) && ((n) → ln_state > ND6_LLININFO_INCOMPLETE))

Definition at line 75 of file nd6.h.

Referenced by `nd6_llinfo_timer()`, `nd6_na_input()`, and `nd6_nud_hint()`.

7.56.1.22 #define ND6_LLININFO_PROBE 4

Definition at line 72 of file nd6.h.

Referenced by `nd6_llinfo_timer()`.

7.56.1.23 #define ND6_LLINFO_REACHABLE 1

Definition at line 69 of file nd6.h.

Referenced by nd6_llinfo_timer(), nd6_na_input(), and nd6_nud_hint().

7.56.1.24 #define ND6_LLINFO_STALE 2

Definition at line 70 of file nd6.h.

Referenced by in6_ifinit(), nd6_cache_lladdr(), nd6_free(), nd6_llinfo_timer(), and nd6_na_input().

7.56.1.25 #define nd6log(x) do { if (nd6_debug) log x; } while ((*CONSTCOND*/ 0)

Definition at line 343 of file nd6.h.

Referenced by defrouter_select(), generate_tmp_ifid(), get_ifid(), icmp6_error(), icmp6_input(), icmp6_redirect_input(), icmp6_reflect(), in6_ifadd(), in6_ifattach(), in6_ifattach_linklocal(), in6_ifattach_loopback(), in6_ifdetach(), in6_tmpifadd(), in6_unlink_ifa(), in6_update_ifa(), ip6_input(), nd6_dad_ns_input(), nd6_dad_start(), nd6_dad_timer(), nd6_lookup(), nd6_na_input(), nd6_na_output(), nd6_ns_input(), nd6_ns_output(), nd6_options(), nd6_prefix_offlink(), nd6_prefix_onlink(), nd6_prelist_add(), nd6_ra_input(), nd6_rs_input(), ni6_input(), pfxlist_onlink_check(), prelist_remove(), and prelist_update().

7.56.1.26 #define ND_COMPUTE_RUNTIME(x)

Value:

```
((MIN_RANDOM_FACTOR * (x >> 10)) + (arc4random() & \
    (MAX_RANDOM_FACTOR - MIN_RANDOM_FACTOR) * (x >> 10))) /1000
```

Definition at line 239 of file nd6.h.

Referenced by nd6_ifattach(), nd6_ioctl(), nd6_ra_input(), and nd6_slowtimo().

7.56.1.27 #define ND_IFINFO(ifp) (((struct in6_ifextra *) (ifp)) → if_afdata[AF_INET6]) → nd_ifinfo)

Definition at line 102 of file nd6.h.

Referenced by icmp6_reflect(), in6_get_tmpifid(), in6_selecthlim(), in6_tmpaddrtimer(), ip6_input(), nd6_dad_duplicated(), nd6_dad_start(), nd6_dad_timer(), nd6_ioctl(), nd6_llinfo_timer(), nd6_na_input(), nd6_nud_hint(), nd6_ra_input(), nd6_setmtu(), and nd6_slowtimo().

7.56.1.28 #define nd_opts_done nd_opt_each.done

Definition at line 377 of file nd6.h.

7.56.1.29 #define nd_opts_last nd_opt_each.last

Definition at line 376 of file nd6.h.

7.56.1.30 #define nd_opts_mtu nd_opt_each.mtu

Definition at line 374 of file nd6.h.

7.56.1.31 #define nd_opts_pi nd_opt_each.pi_beg

Definition at line 371 of file nd6.h.

7.56.1.32 #define nd_opts_pi_end nd_opt_each.pi_end

Definition at line 372 of file nd6.h.

7.56.1.33 #define nd_opts_rh nd_opt_each.rh

Definition at line 373 of file nd6.h.

7.56.1.34 #define nd_opts_search nd_opt_each.search

Definition at line 375 of file nd6.h.

7.56.1.35 #define nd_opts_src_lladdr nd_opt_each.src_lladdr

Definition at line 369 of file nd6.h.

7.56.1.36 #define nd_opts_tgt_lladdr nd_opt_each.tgt_lladdr

Definition at line 370 of file nd6.h.

7.56.1.37 #define ndpr_next ndpr_entry.le_next

Definition at line 289 of file nd6.h.

7.56.1.38 #define ndpr_raf ndpr_flags

Definition at line 291 of file nd6.h.

7.56.1.39 #define ndpr_raf_auto ndpr_flags.autonomous

Definition at line 293 of file nd6.h.

7.56.1.40 #define ndpr_raf_onlink ndpr_flags.onlink

Definition at line 292 of file nd6.h.

7.56.1.41 #define ndpr_raf_router ndpr_flags.router

Definition at line 294 of file nd6.h.

7.56.1.42 #define NDPRF_DETACHED 0x2

Definition at line 219 of file nd6.h.

Referenced by nd6_prefix_offlink(), and pfxlist_onlink_check().

7.56.1.43 #define NDPRF_ONLINK 0x1

Definition at line 218 of file nd6.h.

Referenced by nd6_is_new_addr_neighbor(), nd6_prefix_offlink(), nd6_prefix_onlink(), pfxlist_onlink_check(), prelist_remove(), and prelist_update().

7.56.1.44 #define pfr_next pfr_entry.le_next

Definition at line 324 of file nd6.h.

7.56.1.45 #define PRLSTSIZ 10

Definition at line 121 of file nd6.h.

Referenced by nd6_ioctl().

7.56.1.46 #define prm_raf_auto prm_flags.prf_ra.autonomous

Definition at line 315 of file nd6.h.

7.56.1.47 #define prm_raf_onlink prm_flags.prf_ra.onlink

Definition at line 314 of file nd6.h.

7.56.1.48 #define prm_rrf_decrprefd prm_flags.prf_rr.decrprefd

Definition at line 320 of file nd6.h.

7.56.1.49 #define prm_rrf_decrvalid prm_flags.prf_rr.decrvalid

Definition at line 319 of file nd6.h.

7.56.1.50 #define prm_statef_onlink prm_flags.prf_state.onlink

Definition at line 317 of file nd6.h.

7.56.1.51 #define REACHABLE_TIME 30000

Definition at line 231 of file nd6.h.

Referenced by nd6_ifattach().

7.56.1.52 #define RETRANS_TIMER 1000

Definition at line 232 of file nd6.h.

Referenced by nd6_ifattach().

7.56.1.53 #define RTF_ANNOUNCE RTF_PROTO2

Definition at line 38 of file nd6.h.

Referenced by nd6_ns_input().

7.56.1.54 #define RTR_SOLICITATION_INTERVAL 4

Definition at line 223 of file nd6.h.

7.56.1.55 #define TEMPADDR_REGEN_ADVANCE 5

Definition at line 237 of file nd6.h.

7.56.2 Function Documentation

- 7.56.2.1 `int in6_tmpifadd __P ((const struct in6_ifaddr *, int, int))`
- 7.56.2.2 `int nd6_setdefaultiface __P ((int))`
- 7.56.2.3 `struct nd_prefix* nd6_prefix_lookup __P ((struct nd_prefixctl *))`
- 7.56.2.4 `void rt6_flush __P ((struct in6_addr *, struct ifnet *))`
- 7.56.2.5 `int nd6_prelist_add __P ((struct nd_prefixctl *, struct nd_defrouter *, struct nd_prefix **))`
- 7.56.2.6 `static void defrouter_delreq __P ((struct nd_defrouter *))`
- 7.56.2.7 `static int in6_init_prefix_ltimes __P ((struct nd_prefix *))`
- 7.56.2.8 `void nd6_dad_stop __P ((struct ifaddr *))`
- 7.56.2.9 `void nd6_dad_start __P ((struct ifaddr *, int))`
- 7.56.2.10 `void nd6_ns_output __P ((struct ifnet *, const struct in6_addr *, const struct in6_addr *, struct llinfo_nd6 *, int))`
- 7.56.2.11 `void nd6_na_output __P ((struct ifnet *, const struct in6_addr *, const struct in6_addr *, u_long, int, struct sockaddr *))`
- 7.56.2.12 `void nd6_ra_input __P ((struct mbuf *, int, int))`
- 7.56.2.13 `int nd6_output __P ((struct ifnet *, struct ifnet *, struct mbuf *, struct sockaddr_in6 *, struct rentry *))`
- 7.56.2.14 `struct rentry* nd6_cache_lladdr __P ((struct ifnet *, struct in6_addr *, char *, int, int, int))`
- 7.56.2.15 `int nd6_ioctl __P ((u_long, caddr_t, struct ifnet *))`
- 7.56.2.16 `void nd6_rtrequest __P ((int, struct rentry *, struct rt_addrinfo *))`
- 7.56.2.17 `int nd6_storelladdr __P ((struct ifnet *, struct rentry *, struct mbuf *, struct sockaddr *, u_char *))`
- 7.56.2.18 `void nd6_nud_hint __P ((struct rentry *, struct in6_addr *, int))`
- 7.56.2.19 `void nd6_timer __P ((void *))`
- 7.56.2.20 `void nd6_llinfo_settimer __P ((struct llinfo_nd6 *, long))`
- 7.56.2.21 `struct rentry* nd6_lookup __P ((struct in6_addr *, int, struct ifnet *))`
- 7.56.2.22 `int nd6_options __P ((union nd_opts *))`
- 7.56.2.23 `void nd6_option_init __P ((void *, int, union nd_opts *))`
- 7.56.2.24 `int nd6_is_addr_neighbor __P ((struct sockaddr_in6 *, struct ifnet *))`
- 7.56.2.25 `void nd6_ifdetach __P ((struct nd_ifinfo *))`
- 7.56.2.26 `struct nd_ifinfo* nd6_ifattach __P ((struct ifnet *))`
- 7.56.2.27 `void nd6_init __P ((void))`

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, and `sysctl_ip6_temppltime()`.

7.56.3.2 `u_int32_t ip6_temp_preferred_lifetime`

Definition at line 93 of file `nd6_rtr.c`.

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, `sysctl_ip6_temppltime()`, and `sysctl_ip6_tempvlttime()`.

7.56.3.3 `int ip6_temp_regen_advance`

Definition at line 100 of file `nd6_rtr.c`.

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, and `sysctl_ip6_temppltime()`.

7.56.3.4 `u_int32_t ip6_temp_valid_lifetime`

Definition at line 94 of file `nd6_rtr.c`.

Referenced by `sysctl_ip6_tempvlttime()`.

7.56.3.5 `struct llinfo_nd6 llinfo_nd6`

Definition at line 102 of file `nd6.c`.

Referenced by `nd6_free()`, `nd6_llinfo_timer()`, `nd6_lookup()`, `nd6_nud_hint()`, `nd6_purge()`, and `nd6_rtrequest()`.

7.56.3.6 `int nd6_debug`

Definition at line 96 of file `nd6.c`.

7.56.3.7 `int nd6_defindex`

Definition at line 88 of file `nd6_rtr.c`.

Referenced by `nd6_ioctl()`, and `nd6_is_new_addr_neighbor()`.

7.56.3.8 `int nd6_delay`

Definition at line 81 of file `nd6.c`.

7.56.3.9 `int nd6_gctimer`

Definition at line 85 of file `nd6.c`.

Referenced by `nd6_cache_lladdr()`, `nd6_free()`, `nd6_llinfo_timer()`, and `nd6_na_input()`.

7.56.3.10 `int nd6_maxnudhint`

Definition at line 90 of file `nd6.c`.

Referenced by `nd6_nud_hint()`.

7.56.3.11 `int` [nd6_mmaxtries](#)

Definition at line 83 of file `nd6.c`.

Referenced by `nd6_llinfo_timer()`.

7.56.3.12 `int` [nd6_prune](#)

Definition at line 80 of file `nd6.c`.

Referenced by `nd6_timer()`.

7.56.3.13 `struct callout` [nd6_timer_ch](#)

Definition at line 119 of file `nd6.c`.

Referenced by `ip6_init2()`.

7.56.3.14 `int` [nd6_umaxtries](#)

Definition at line 82 of file `nd6.c`.

Referenced by `nd6_llinfo_timer()`.

7.56.3.15 `int` [nd6_uselookback](#)

Definition at line 84 of file `nd6.c`.

7.56.3.16 `struct nd_drhead` [nd_defrouter](#)

Definition at line 103 of file `nd6.c`.

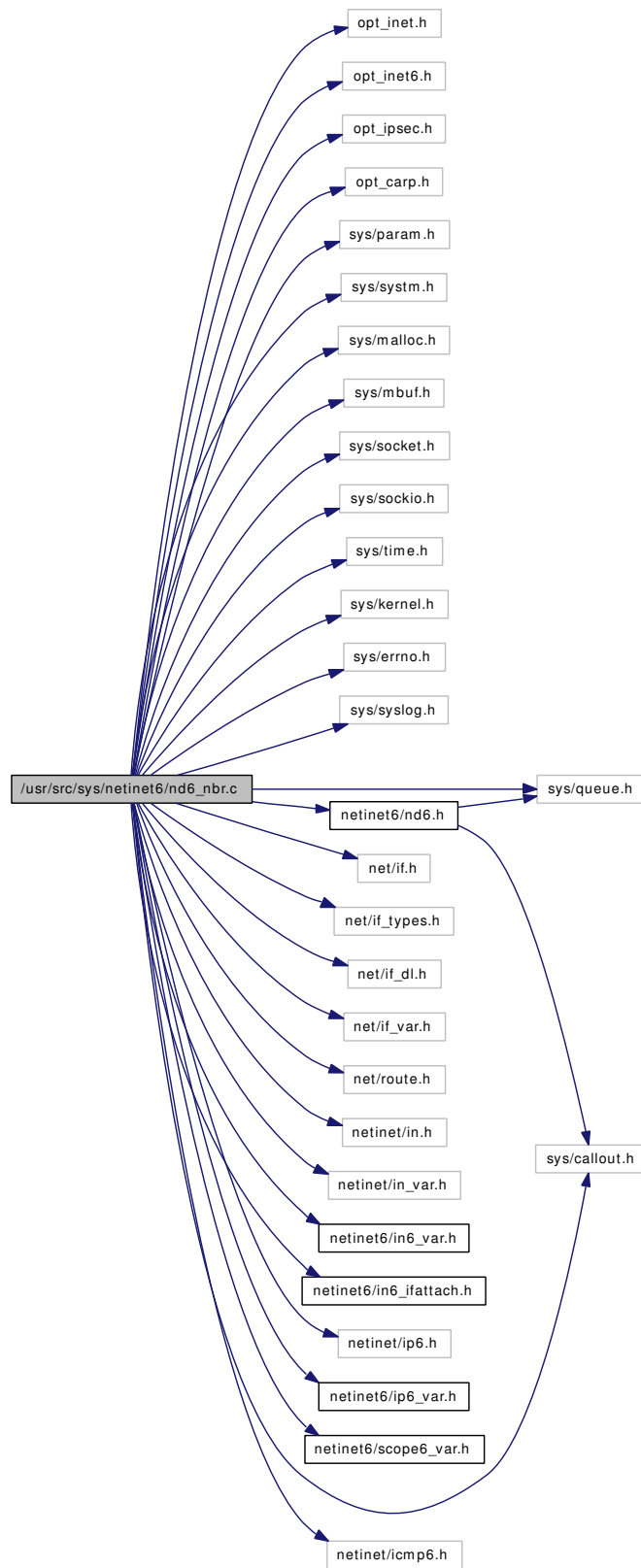
7.56.3.17 `struct nd_prhead` [nd_prefix](#)

Definition at line 104 of file `nd6.c`.

7.57 /usr/src/sys/netinet6/nd6_nbr.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include "opt_carp.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/sockio.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/queue.h>
#include <sys/callout.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/if_dl.h>
#include <net/if_var.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet6/in6_var.h>
#include <netinet6/in6_ifattach.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/scope6_var.h>
#include <netinet6/nd6.h>
#include <netinet/icmp6.h>
```

Include dependency graph for nd6_nbr.c:



Data Structures

- struct [dadq](#)

Defines

- #define [SDL\(s\)](#) ((struct sockaddr_dl *)s)

Functions

- static struct [dadq](#) *nd6_dad_find __P ((struct ifaddr *))
- static void nd6_dad_starttimer __P ((struct [dadq](#) *, int))
- static void nd6_dad_stoptimer __P ((struct [dadq](#) *))
- static void nd6_dad_ns_output __P ((struct [dadq](#) *, struct ifaddr *))
- void [nd6_ns_input](#) (struct mbuf *m, int off, int icmp6len)
- void [nd6_ns_output](#) (struct ifnet *ifp, const struct [in6_addr](#) *daddr6, const struct [in6_addr](#) *taddr6, struct [llinfo_nd6](#) *ln, int dad)
- void [nd6_na_input](#) (struct mbuf *m, int off, int icmp6len)
- void [nd6_na_output](#) (struct ifnet *ifp, const struct [in6_addr](#) *daddr6_0, const struct [in6_addr](#) *taddr6, u_long flags, int ttladdr, struct sockaddr *sdl0)
- caddr_t [nd6_ifptomac](#) (struct ifnet *ifp)
- [TAILQ_HEAD](#) (dadq_head, [dadq](#))
- static struct [dadq](#) * [nd6_dad_find](#) (struct ifaddr *ifa)
- static void [nd6_dad_starttimer](#) (struct [dadq](#) *dp, int ticks)
- static void [nd6_dad_stoptimer](#) (struct [dadq](#) *dp)
- void [nd6_dad_start](#) (struct ifaddr *ifa, int delay)
- void [nd6_dad_stop](#) (struct ifaddr *ifa)
- static void [nd6_dad_timer](#) (struct ifaddr *ifa)
- void [nd6_dad_duplicated](#) (struct ifaddr *ifa)
- static void [nd6_dad_ns_output](#) (struct [dadq](#) *dp, struct ifaddr *ifa)
- static void [nd6_dad_ns_input](#) (struct ifaddr *ifa)
- static void [nd6_dad_na_input](#) (struct ifaddr *ifa)

Variables

- static int [dad_ignore_ns](#) = 0
- static int [dad_maxtry](#) = 15
- static struct dadq_head [dadq](#)
- static int [dad_init](#) = 0

7.57.1 Define Documentation

7.57.1.1 #define [SDL\(s\)](#) ((struct sockaddr_dl *)s)

Definition at line 71 of file nd6_nbr.c.

7.57.2 Function Documentation

7.57.2.1 `static void nd6_dad_ns_output __P ((struct dadq *, struct ifaddr *))` [static]

7.57.2.2 `static void nd6_dad_stoptimer __P ((struct dadq *))` [static]

7.57.2.3 `static void nd6_dad_starttimer __P ((struct dadq *, int))` [static]

7.57.2.4 `static struct dadq* nd6_dad_find __P ((struct ifaddr *))` [static]

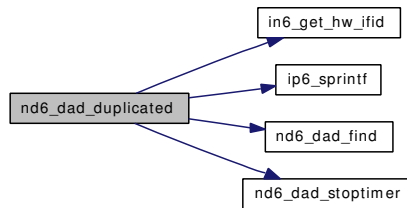
7.57.2.5 `void nd6_dad_duplicated (struct ifaddr * ifa)`

Definition at line 1315 of file nd6_nbr.c.

References `in6_ifaddr::ia6_flags`, `in6_ifaddr::ia_addr`, `IN6_ARE_ADDR_EQUAL`, `in6_get_hw_ifid()`, `IN6_IFF_DUPLICATED`, `IN6_IFF_TENTATIVE`, `IN6_IS_ADDR_LINKLOCAL`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `nd6_dad_find()`, `nd6_dad_stoptimer()`, `ND6_IFF_IFDISABLED`, `ND_IFINFO`, and `sockaddr_in6::sin6_addr`.

Referenced by `nd6_dad_na_input()`, `nd6_dad_ns_input()`, and `nd6_dad_timer()`.

Here is the call graph for this function:



7.57.2.6 `static struct dadq* nd6_dad_find (struct ifaddr * ifa)` [static]

Definition at line 1070 of file nd6_nbr.c.

Referenced by `nd6_dad_duplicated()`, `nd6_dad_na_input()`, `nd6_dad_ns_input()`, `nd6_dad_start()`, `nd6_dad_stop()`, and `nd6_dad_timer()`.

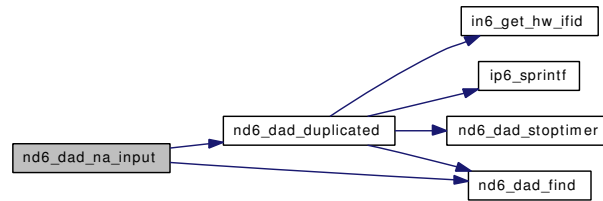
7.57.2.7 `static void nd6_dad_na_input (struct ifaddr * ifa)` [static]

Definition at line 1458 of file nd6_nbr.c.

References `nd6_dad_duplicated()`, and `nd6_dad_find()`.

Referenced by `nd6_na_input()`.

Here is the call graph for this function:



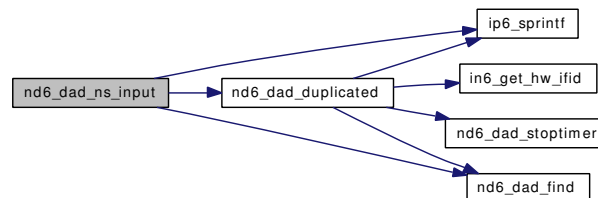
7.57.2.8 static void nd6_dad_ns_input(struct ifaddr *ifa) [static]

Definition at line 1407 of file nd6_nbr.c.

References in6_ifaddr::ia_addr, INET6_ADDRSTRLEN, ip6_sprintf(), nd6_dad_duplicated(), nd6_dad_find(), nd6log, and sockaddr_in6::sin6_addr.

Referenced by nd6_ns_input().

Here is the call graph for this function:



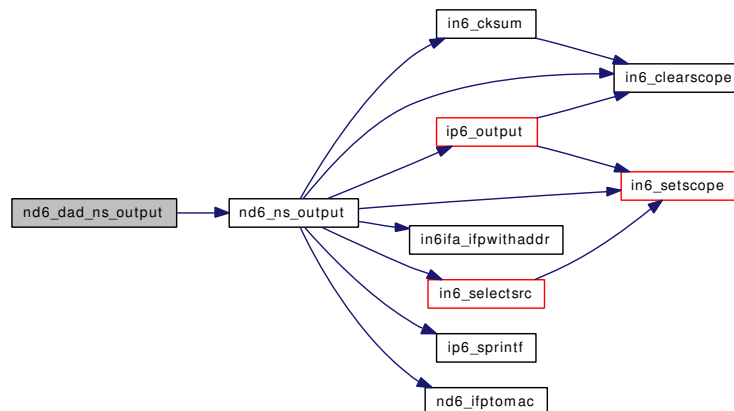
7.57.2.9 static void nd6_dad_ns_output(struct dadq *dp, struct ifaddr *ifa) [static]

Definition at line 1387 of file nd6_nbr.c.

References in6_ifaddr::ia_addr, nd6_ns_output(), and sockaddr_in6::sin6_addr.

Referenced by nd6_dad_start(), and nd6_dad_timer().

Here is the call graph for this function:



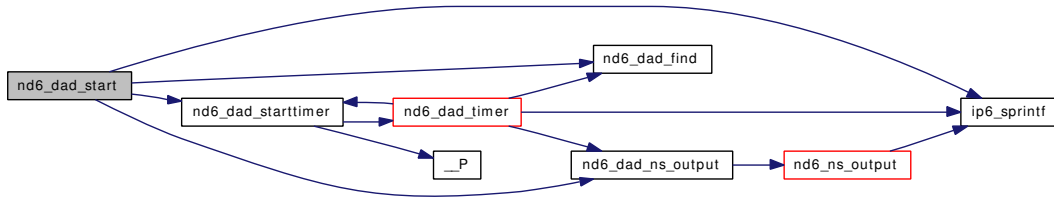
7.57.2.10 void nd6_dad_start (struct ifaddr * ifa, int delay)

Definition at line 1104 of file nd6_nbr.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia_addr, IN6_IFF_ANYCAST, IN6_IFF_TENTATIVE, INET6_ADDRSTRLEN, ip6_dad_count, ip6_sprintf(), nd6_dad_find(), nd6_dad_ns_output(), nd6_dad_starttimer(), nd6log, ND_IFINFO, and sockaddr_in6::sin6_addr.

Referenced by in6_if_up(), and in6_update_ifa().

Here is the call graph for this function:



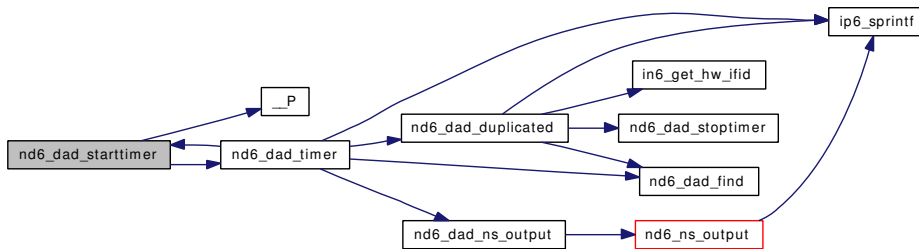
7.57.2.11 static void nd6_dad_starttimer (struct dadq * dp, int ticks) [static]

Definition at line 1083 of file nd6_nbr.c.

References __P(), and nd6_dad_timer().

Referenced by nd6_dad_start(), and nd6_dad_timer().

Here is the call graph for this function:



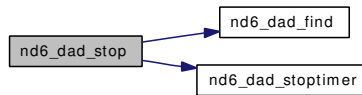
7.57.2.12 void nd6_dad_stop (struct ifaddr * ifa)

Definition at line 1188 of file nd6_nbr.c.

References nd6_dad_find(), and nd6_dad_stoptimer().

Referenced by in6_purgeaddr().

Here is the call graph for this function:



7.57.2.13 static void nd6_dad_stoptimer (struct dadq * dp) [static]

Definition at line 1093 of file nd6_nbr.c.

Referenced by nd6_dad_duplicated(), and nd6_dad_stop().

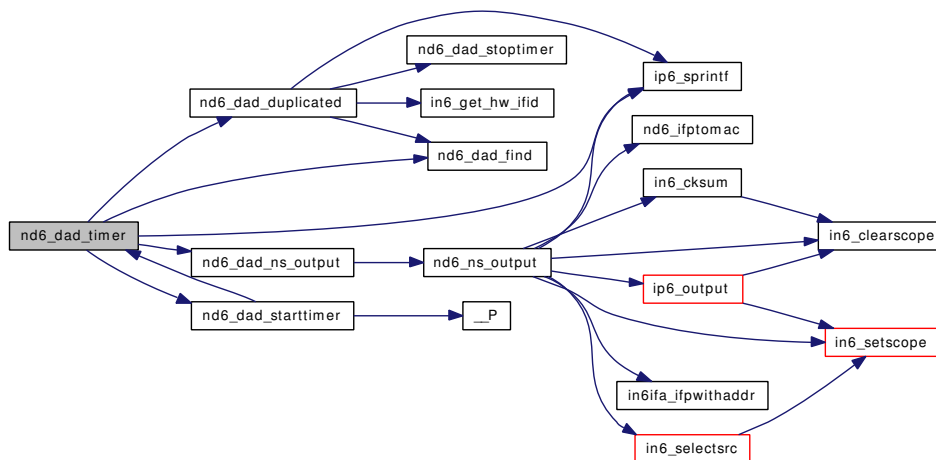
7.57.2.14 static void nd6_dad_timer (struct ifaddr * ifa) [static]

Definition at line 1210 of file nd6_nbr.c.

References in6_ifaddr::ia6_flags, in6_ifaddr::ia_addr, IN6_IFF_DUPLICATED, IN6_IFF_TENTATIVE, INET6_ADDRSTRLEN, ip6_sprintf(), nd6_dad_duplicated(), nd6_dad_find(), nd6_dad_ns_output(), nd6_dad_starttimer(), nd6log, ND_IFINFO, and sockaddr_in6::sin6_addr.

Referenced by nd6_dad_starttimer().

Here is the call graph for this function:



7.57.2.15 caddr_t nd6_ifptomac (struct ifnet * ifp)

Definition at line 1029 of file nd6_nbr.c.

Referenced by nd6_na_output(), and nd6_ns_output().

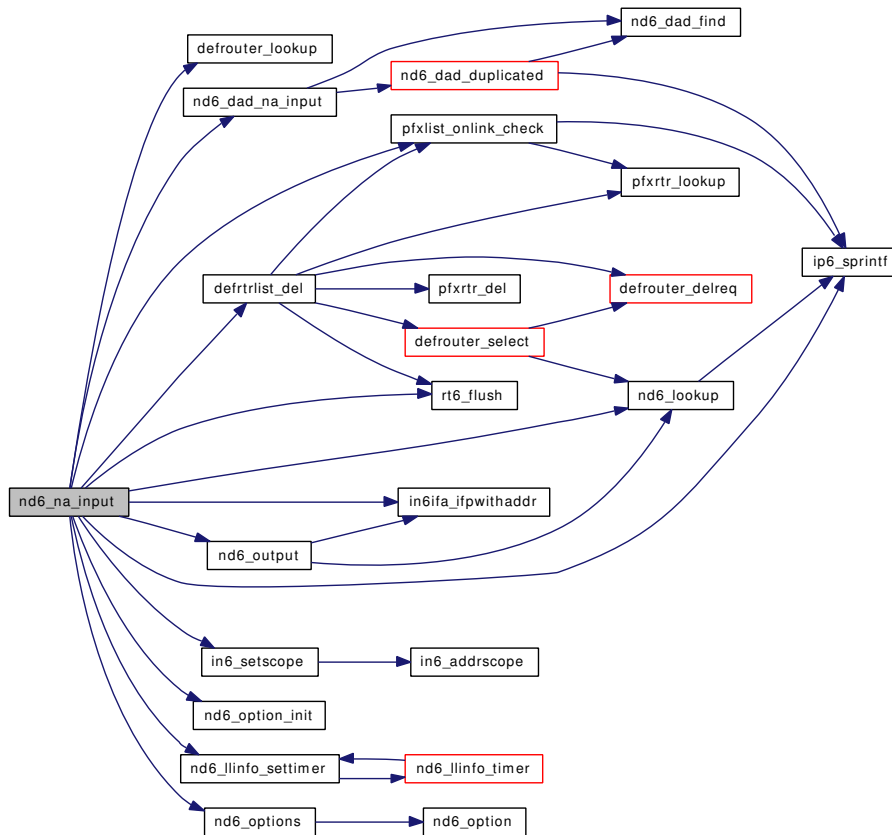
7.57.2.16 void nd6_na_input (struct mbuf * m, int off, int icmp6len)

Definition at line 550 of file nd6_nbr.c.

References defrouter_lookup(), defrtrlist_del(), icmp6stat, IN6_IFF_TENTATIVE, IN6_IS_ADDR_MULTICAST, in6_setscope(), in6ifa_ifpwithaddr(), INET6_ADDRSTRLEN, ip6_forwarding, ip6_sprintf(), llinfo_nd6::ln_asked, llinfo_nd6::ln_byhint, llinfo_nd6::ln_hold, llinfo_nd6::ln_router, llinfo_nd6::ln_state, nd6_dad_na_input(), nd6_gctimer, ND6_LLINFO_INCOMPLETE, ND6_LLINFO_PERMANENT, ND6_LLINFO_REACHABLE, nd6_llinfo_settimer(), ND6_LLINFO_STALE, nd6_lookup(), nd6_option_init(), nd6_options(), nd6_output(), nd6log, ND_IFINFO, pfxlist_onlink_check(), rt6_flush(), SDL, and sockaddr_in6::sin6_addr.

Referenced by icmp6_input().

Here is the call graph for this function:



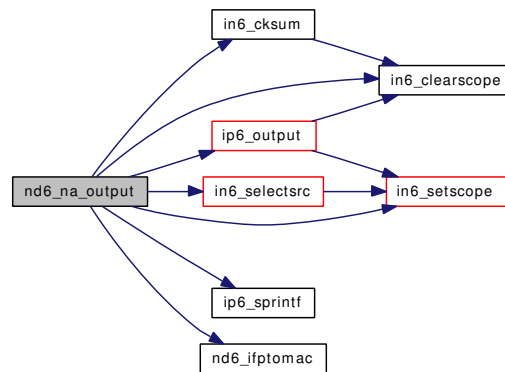
7.57.2.17 void nd6_na_output (struct ifnet * ifp, const struct in6_addr * daddr6_0, const struct in6_addr * taddr6, u_long flags, int tlladdr, struct sockaddr * sdl0)

Definition at line 860 of file nd6_nbr.c.

References icmp6stat, ip6_moptions::im6o_multicast_hlim, ip6_moptions::im6o_multicast_ifp, ip6_moptions::im6o_multicast_loop, in6_cksum(), in6_clearscope(), IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, in6_selectsrc(), in6_setscope(), INET6_ADDRSTRLEN, ip6_output(), ip6_sprintf(), IPV6_ADDR_INT16_MLL, IPV6_ADDR_INT32_ONE, nd6_ifptomac(), nd6log, sockaddr_in6::sin6_addr, sockaddr_in6::sin6_family, and sockaddr_in6::sin6_len.

Referenced by nd6_ns_input().

Here is the call graph for this function:



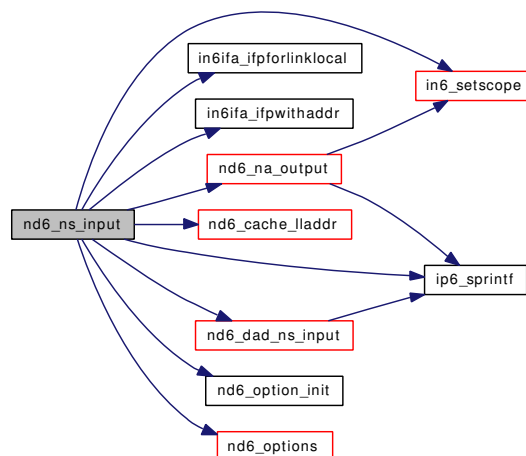
7.57.2.18 void nd6_ns_input (struct mbuf * m, int off, int icmp6len)

Definition at line 92 of file nd6_nbr.c.

References in6_ifaddr::ia6_flags, icmp6stat, IFA_INET6, IN6_ARE_ADDR_EQUAL, IN6_IFF_-ANYCAST, IN6_IFF_DUPLICATED, IN6_IFF_NOTREADY, IN6_IFF_TENTATIVE, IN6_IS_-ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, in6_setscope(), in6addr_linklocal_allnodes, in6ifa_ifpforlinklocal(), in6ifa_ifpwithaddr(), INET6_ADDRSTRLEN, ip6_forwarding, ip6_sprintf(), IPV6_ADDR_INT16_MLL, IPV6_ADDR_INT32_ONE, nd6_cache_lladdr(), nd6_dad_ns_input(), nd6_na_output(), nd6_option_init(), nd6_options(), nd6log, RTF_ANNOUNCE, and SDL.

Referenced by icmp6_input().

Here is the call graph for this function:



7.57.2.19 void nd6_ns_output (struct ifnet * ifp, const struct in6_addr * daddr6, const struct in6_addr * taddr6, struct linfo_nd6 * ln, int dad)

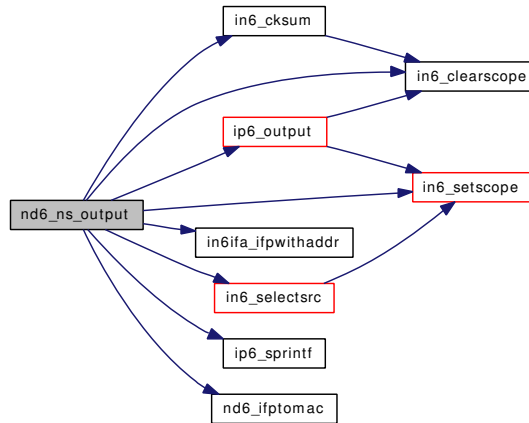
Definition at line 341 of file nd6_nbr.c.

References icmp6stat, ip6_moptions::im6o_multicast_hlim, ip6_moptions::im6o_multicast_ifp, ip6_moptions::im6o_multicast_loop, in6_cksum(), in6_clearscope(), IN6_IS_ADDR_MULTICAST, in6_-

selectsrc(), in6_setscope(), in6ifa_ifpwithaddr(), INET6_ADDRSTRLEN, ip6_output(), ip6_sprintf(), IPV6_ADDR_INT16_MLL, IPV6_ADDR_INT32_ONE, IPV6_UNSPECSRC, llinfo_nd6::ln_hold, nd6_ifptomac(), and nd6log.

Referenced by nd6_dad_ns_output(), and nd6_llinfo_timer().

Here is the call graph for this function:



7.57.2.20 TAILQ_HEAD (dadq_head, dadq)

7.57.3 Variable Documentation

7.57.3.1 int dad_ignore_ns = 0 [static]

Definition at line 82 of file nd6_nbr.c.

7.57.3.2 int dad_init = 0 [static]

Definition at line 1067 of file nd6_nbr.c.

7.57.3.3 int dad_maxtry = 15 [static]

Definition at line 83 of file nd6_nbr.c.

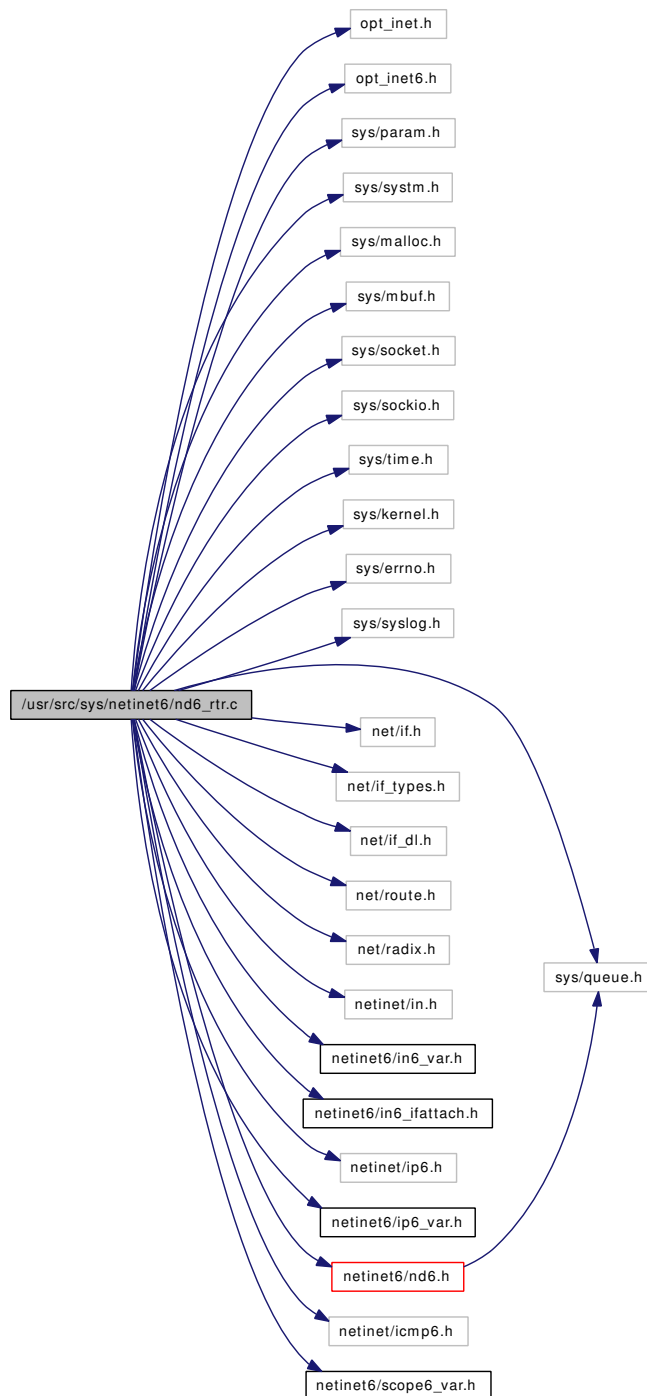
7.57.3.4 struct dadq_head dadq [static]

Definition at line 1066 of file nd6_nbr.c.

7.58 /usr/src/sys/netinet6/nd6_rtr.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/system.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/sockio.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <sys/queue.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/if_dl.h>
#include <net/route.h>
#include <net/radix.h>
#include <netinet/in.h>
#include <netinet6/in6_var.h>
#include <netinet6/in6_ifattach.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/nd6.h>
#include <netinet/icmp6.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for nd6_rtr.c:



Defines

- #define [SDL\(s\)](#) ((struct sockaddr_dl *)s)
- #define [RTPREF_HIGH](#) 1
- #define [RTPREF_MEDIUM](#) 0
- #define [RTPREF_LOW](#) (-1)
- #define [RTPREF_RESERVED](#) (-2)

- #define RTPREF_INVALID (-3)
- #define TWOHOUR (120*60)
- #define SIN6(s) ((struct sockaddr_in6 *)s)

Functions

- static int rtpref __P ((struct nd_defrouter *))
- static int prelist_update __P ((struct nd_prefixctl *, struct nd_defrouter *, struct mbuf *, int))
- static struct in6_ifaddr *in6_ifadd __P ((struct nd_prefixctl *, int))
- static struct nd_pfxrouter *pfxrtr_lookup __P ((struct nd_prefix *, struct nd_defrouter *))
- static void pfxrtr_add __P ((struct nd_prefix *, struct nd_defrouter **))
- static void pfxrtr_del __P ((struct nd_pfxrouter **))
- static struct nd_pfxrouter *find_pfxlist_reachable_router __P ((struct nd_prefix *))
- static void nd6_rtmsg __P ((int, struct rtentry *))
- static void in6_init_address_ltimes __P ((struct nd_prefix *, struct in6_addrlifetime *))
- static int rt6_deleteroute __P ((struct radix_node *, void **))
- void nd6_rs_input (struct mbuf *m, int off, int icmp6len)
- void nd6_ra_input (struct mbuf *m, int off, int icmp6len)
- static void nd6_rtmsg (int cmd, struct rtentry *rt)
- void defrouter_addrq (struct nd_defrouter *new)
- nd_defrouter * defrouter_lookup (struct in6_addr *addr, struct ifnet *ifp)
- static void defrouter_delreq (struct nd_defrouter *dr)
- void defrouter_reset ()
- void defrtrlist_del (struct nd_defrouter *dr)
- void defrouter_select ()
- static int rtpref (struct nd_defrouter *dr)
- static struct nd_defrouter * defrtrlist_update (struct nd_defrouter *new)
- static struct nd_pfxrouter * pfxrtr_lookup (struct nd_prefix *pr, struct nd_defrouter *dr)
- static void pfxrtr_add (struct nd_prefix *pr, struct nd_defrouter *dr)
- static void pfxrtr_del (struct nd_pfxrouter *pfr)
- nd_prefix * nd6_prefix_lookup (struct nd_prefixctl *key)
- int nd6_prelist_add (struct nd_prefixctl *pr, struct nd_defrouter *dr, struct nd_prefix **newp)
- void prelist_remove (struct nd_prefix *pr)
- static int prelist_update (struct nd_prefixctl *new, struct nd_defrouter *dr, struct mbuf *m, int mcast)
- static struct nd_pfxrouter * find_pfxlist_reachable_router (struct nd_prefix *pr)
- void pfxlist_onlink_check ()
- int nd6_prefix_onlink (struct nd_prefix *pr)
- int nd6_prefix_offlink (struct nd_prefix *pr)
- static struct in6_ifaddr * in6_ifadd (struct nd_prefixctl *pr, int mcast)
- int in6_tmpifadd (struct in6_ifaddr *ia0, int forcegen, int delay) const
- static int in6_init_prefix_ltimes (struct nd_prefix *ndpr)
- static void in6_init_address_ltimes (struct nd_prefix *new, struct in6_addrlifetime *lt6)
- void rt6_flush (struct in6_addr *gateway, struct ifnet *ifp)
- static int rt6_deleteroute (struct radix_node *rn, void *arg)
- int nd6_setdefaultiface (int ifindex)

Variables

- int [nd6_recalc_reachtm_interval](#)
- static struct ifnet * [nd6_defifp](#)
- int [nd6_defifindex](#)
- int [ip6_use_tempaddr](#) = 0
- int [ip6_desync_factor](#)
- u_int32_t [ip6_temp_preferred_lifetime](#) = DEF_TEMP_PREFERRED_LIFETIME
- u_int32_t [ip6_temp_valid_lifetime](#) = DEF_TEMP_VALID_LIFETIME
- int [ip6_temp_regen_advance](#) = TEMPADDR_REGEN_ADVANCE

7.58.1 Define Documentation

7.58.1.1 #define RTPREF_HIGH 1

Definition at line 103 of file [nd6_rtr.c](#).

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

7.58.1.2 #define RTPREF_INVALID (-3)

Definition at line 107 of file [nd6_rtr.c](#).

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

7.58.1.3 #define RTPREF_LOW (-1)

Definition at line 105 of file [nd6_rtr.c](#).

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

7.58.1.4 #define RTPREF_MEDIUM 0

Definition at line 104 of file [nd6_rtr.c](#).

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

7.58.1.5 #define RTPREF_RESERVED (-2)

Definition at line 106 of file [nd6_rtr.c](#).

7.58.1.6 #define SDL(s) ((struct sockaddr_dl *)s)

Definition at line 63 of file [nd6_rtr.c](#).

7.58.1.7 #define SIN6(s) ((struct sockaddr_in6 *)s)

7.58.1.8 #define TWOHOUR (120*60)

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

7.58.2 Function Documentation

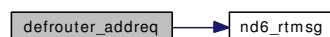
- 7.58.2.1** `static int rt6_deleteroute __P((struct radix_node *, void *))` [static]
- 7.58.2.2** `static void in6_init_address_ltimes __P((struct nd_prefix *, struct in6_addrlifetime *))` [static]
- 7.58.2.3** `static void nd6_rtmsg __P((int, struct rtentry *))` [static]
- 7.58.2.4** `static struct nd_pfxrouter* find_pfxlist_reachable_router __P((struct nd_prefix *))` [static]
- 7.58.2.5** `static void pfxrtr_del __P((struct nd_pfxrouter *))` [static]
- 7.58.2.6** `static void pfxrtr_add __P((struct nd_prefix *, struct nd_defrouter *))` [static]
- 7.58.2.7** `static struct nd_pfxrouter* pfxrtr_lookup __P((struct nd_prefix *, struct nd_defrouter *))` [static]
- 7.58.2.8** `static struct in6_ifaddr* in6_ifadd __P((struct nd_prefixctl *, int))` [static]
- 7.58.2.9** `static int prelist_update __P((struct nd_prefixctl *, struct nd_defrouter *, struct mbuf *, int))` [static]
- 7.58.2.10** `static int rtpref __P((struct nd_defrouter *))` [static]
- 7.58.2.11** `void defrouter_addreq (struct nd_defrouter * new)`

Definition at line 454 of file nd6_rtr.c.

References nd6_rtmsg().

Referenced by defrouter_select().

Here is the call graph for this function:



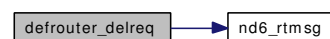
- 7.58.2.12** `static void defrouter_delreq (struct nd_defrouter * dr)` [static]

Definition at line 509 of file nd6_rtr.c.

References nd6_rtmsg().

Referenced by defrouter_reset(), defrouter_select(), and defrtrlist_del().

Here is the call graph for this function:



7.58.2.13 struct `nd_defrouter*` `defrouter_lookup` (struct `in6_addr *` *addr*, struct `ifnet *` *ifp*)

Definition at line 488 of file `nd6_rtr.c`.

References `nd_defrouter::ifp`, `IN6_ARE_ADDR_EQUAL`, and `nd_defrouter::rtaddr`.

Referenced by `defrtrlist_update()`, `nd6_free()`, and `nd6_na_input()`.

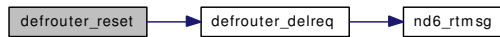
7.58.2.14 void `defrouter_reset` ()

Definition at line 539 of file `nd6_rtr.c`.

References `defrouter_delreq()`.

Referenced by `nd6_ioctl()`.

Here is the call graph for this function:



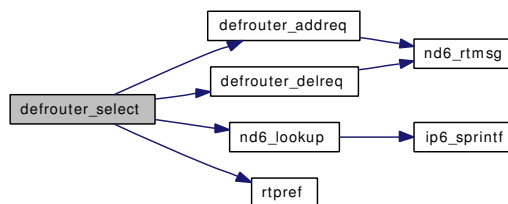
7.58.2.15 void `defrouter_select` ()

Definition at line 616 of file `nd6_rtr.c`.

References `defrouter_addreq()`, `defrouter_delreq()`, `nd_defrouter::ifp`, `nd_defrouter::installed`, `ip6_accept_rtadv`, `ip6_forwarding`, `ND6_IS_LLINFO_PROBREACH`, `nd6_lookup()`, `nd6log`, `nd_defrouter::rtaddr`, and `rtpref()`.

Referenced by `defrtrlist_del()`, `defrtrlist_update()`, `nd6_cache_lladdr()`, `nd6_free()`, and `nd6_ioctl()`.

Here is the call graph for this function:



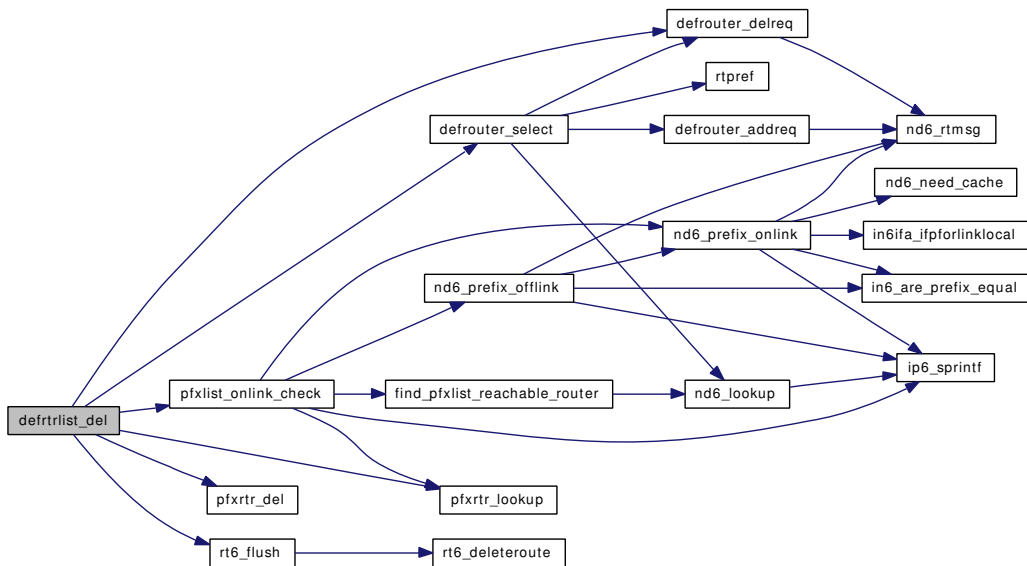
7.58.2.16 void `defrtrlist_del` (struct `nd_defrouter *` *dr*)

Definition at line 554 of file `nd6_rtr.c`.

References `defrouter_delreq()`, `defrouter_select()`, `ip6_accept_rtadv`, `ip6_forwarding`, `pfxlist_onlink_check()`, `pfxrtr_del()`, `pfxrtr_lookup()`, and `rt6_flush()`.

Referenced by `defrtrlist_update()`, `nd6_ioctl()`, `nd6_na_input()`, `nd6_purge()`, and `nd6_timer()`.

Here is the call graph for this function:



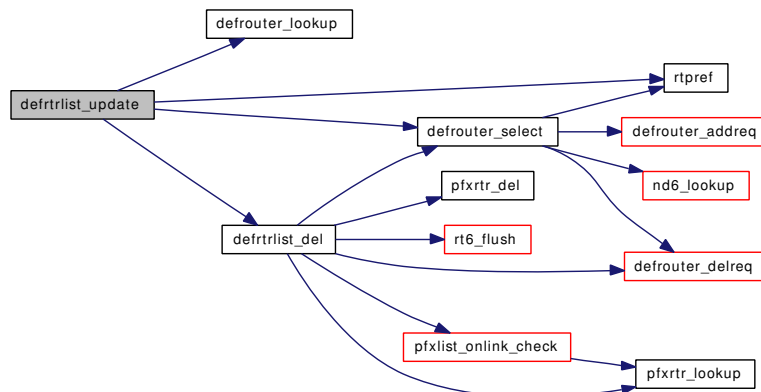
7.58.2.17 static struct **nd_defrouter*** defrtrlist_update (struct **nd_defrouter** * new) [static]

Definition at line 732 of file nd6_rtr.c.

References defrouter_lookup(), defrouter_select(), defrtrlist_del(), nd_defrouter::expire, nd_defrouter::flags, nd_defrouter::rtlifetime, and rtpref().

Referenced by nd6_ra_input().

Here is the call graph for this function:



7.58.2.18 static struct **nd_pfxrouter*** find_pfxlist_reachable_router (struct **nd_prefix** * pr) [static]

Definition at line 1328 of file nd6_rtr.c.

References nd_defrouter::ifp, ND6_IS_LLINFO_PROBREACH, nd6_lookup(), nd_pfxrouter::router, and nd_defrouter::rtaddr.

Referenced by `pfxlist_onlink_check()`.

Here is the call graph for this function:



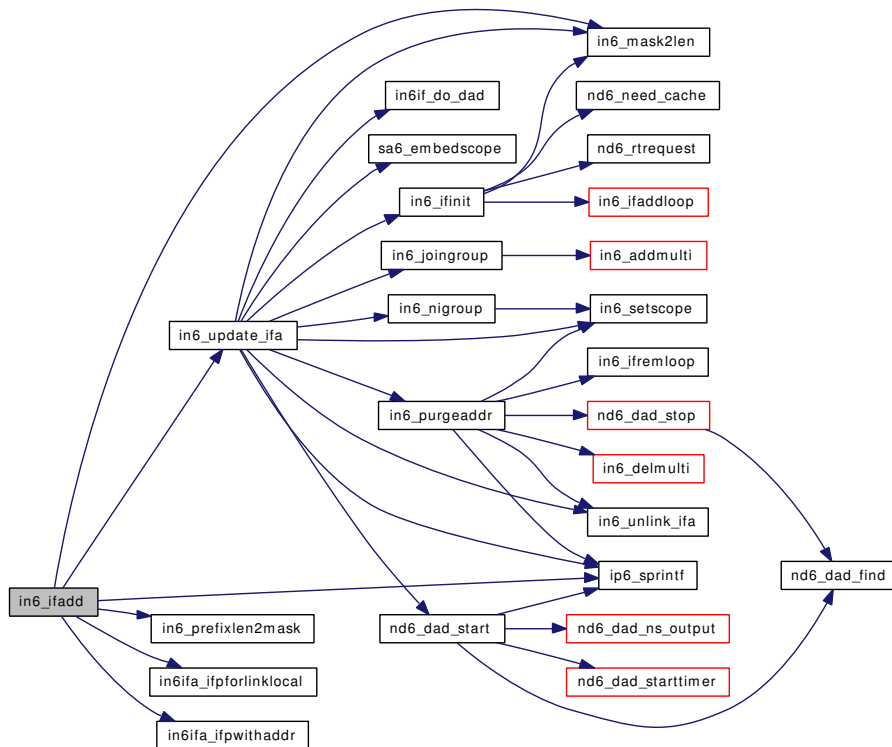
7.58.2.19 `static struct in6_ifaddr* in6_ifadd(struct nd_prefixctl *pr, int mcast)` [static]

Definition at line 1746 of file `nd6_rtr.c`.

References `in6_ifaddr::ia_addr`, `in6_ifaddr::ia_prefixmask`, `IN6_IFAUPDATE_DADDELAY`, `in6_mask2len()`, `in6_prefixlen2mask()`, `in6_update_ifa()`, `in6ifa_ifpforlinklocal()`, `in6ifa_ifpwithaddr()`, `INET6_ADDRSTRLEN`, `ip6_sprintf()`, `nd6log`, and `sockaddr_in6::sin6_addr`.

Referenced by `prelist_update()`.

Here is the call graph for this function:



7.58.2.20 `static void in6_init_address_ltimes(struct nd_prefix *new, struct in6_addrlifetime *lt6)` [static]

Definition at line 2017 of file `nd6_rtr.c`.

References `in6_addrlifetime::ia6t_expire`, `in6_addrlifetime::ia6t_pltime`, `in6_addrlifetime::ia6t_preferred`, `in6_addrlifetime::ia6t_vltime`, and `ND6_INFINITE_LIFETIME`.

Referenced by prelist_update().

7.58.2.21 static int in6_init_prefix_ltimes (struct nd_prefix * ndpr) [static]

Definition at line 2002 of file nd6_rtr.c.

References ND6_INFINITE_LIFETIME, nd_prefix::ndpr_expire, nd_prefix::ndpr_pltime, nd_prefix::ndpr_preferred, and nd_prefix::ndpr_vltime.

Referenced by nd6_prelist_add(), and prelist_update().

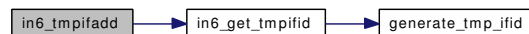
7.58.2.22 int in6_tmpifadd (struct in6_ifaddr * ia0, int forcegen, int delay) const

Definition at line 1873 of file nd6_rtr.c.

References in6_get_tmpifid(), and nd6log.

Referenced by in6_control(), prelist_update(), and regen_tmpaddr().

Here is the call graph for this function:



7.58.2.23 struct nd_prefix* nd6_prefix_lookup (struct nd_prefixctl * key)

Definition at line 859 of file nd6_rtr.c.

References in6_are_prefix_equal(), nd_prefix::ndpr_ifp, nd_prefix::ndpr_plen, nd_prefix::ndpr_prefix, and sockaddr_in6::sin6_addr.

Referenced by in6_control(), in6_ifattach_linklocal(), and prelist_update().

Here is the call graph for this function:



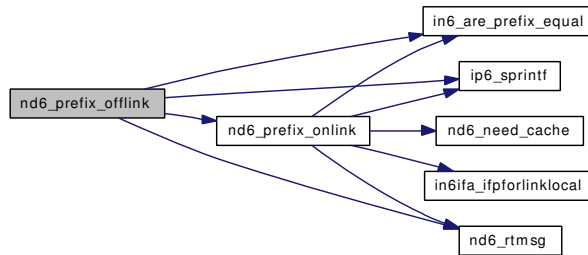
7.58.2.24 int nd6_prefix_offlink (struct nd_prefix * pr)

Definition at line 1654 of file nd6_rtr.c.

References in6_are_prefix_equal(), INET6_ADDRSTRLEN, ip6_sprintf(), nd6_prefix_onlink(), nd6_rtmmsg(), nd6log, nd_prefix::ndpr_ifp, nd_prefix::ndpr_plen, nd_prefix::ndpr_prefix, nd_prefix::ndpr_stateflags, NDPRF_DETACHED, NDPRF_ONLINK, and sockaddr_in6::sin6_addr.

Referenced by pfxlist_onlink_check(), and prelist_remove().

Here is the call graph for this function:



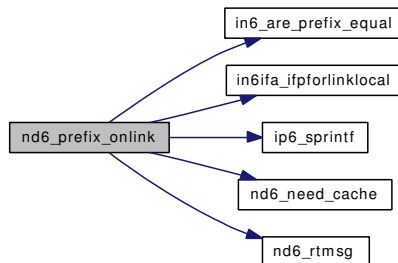
7.58.2.25 int nd6_prefix_onlink (struct nd_prefix * pr)

Definition at line 1540 of file nd6_rtr.c.

References in6_are_prefix_equal(), IN6_IFF_ANYCAST, IN6_IFF_NOTREADY, in6ifa_ifpforlinklocal(), INET6_ADDRSTRLEN, ip6_sprintf(), nd6_need_cache(), nd6_rtmsg(), nd6log, nd_prefix::ndpr_plen, nd_prefix::ndpr_prefix, nd_prefix::ndpr_stateflags, NDPRF_ONLINK, and sockaddr_in6::sin6_addr.

Referenced by nd6_prefix_offlink(), nd6_prelist_add(), pfxlist_onlink_check(), and prelist_update().

Here is the call graph for this function:



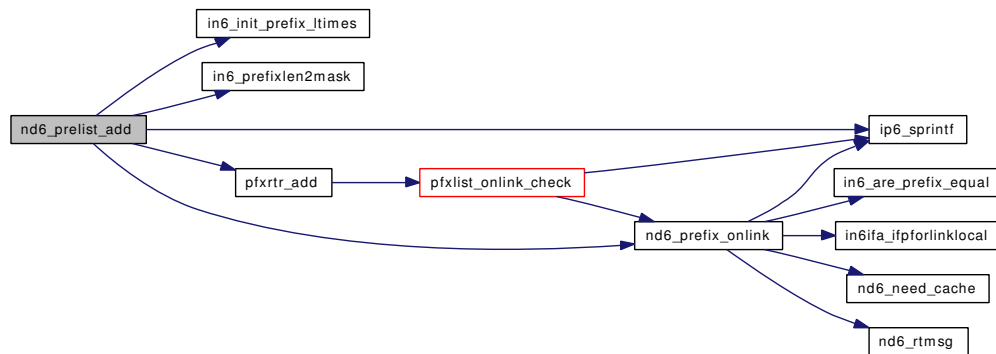
7.58.2.26 int nd6_prelist_add (struct nd_prefixctl * pr, struct nd_defrouter * dr, struct nd_prefix ** newp)

Definition at line 877 of file nd6_rtr.c.

References in6_init_prefix_ltimes(), in6_prefixlen2mask(), INET6_ADDRSTRLEN, ip6_sprintf(), nd6_prefix_onlink(), nd6log, and pfxrtr_add().

Referenced by in6_control(), in6_ifattach_linklocal(), and prelist_update().

Here is the call graph for this function:



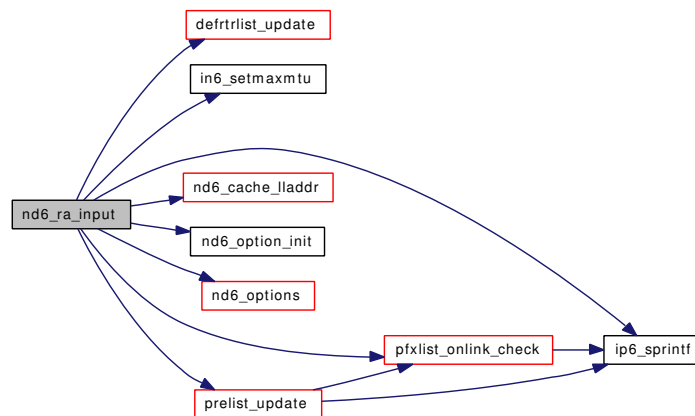
7.58.2.27 void nd6_ra_input (struct mbuf * m, int off, int icmp6len)

Definition at line 203 of file nd6_rtr.c.

References nd_ifinfo::basereachable, nd_ifinfo::chlim, defrtrlist_update(), nd_ifinfo::flags, icmp6stat, IN6_IS_ADDR_LINKLOCAL, IN6_IS_ADDR_MULTICAST, in6_setmaxmtu(), INET6_ADDRSTRLEN, ip6_accept_rtadv, ip6_sprintf(), nd_ifinfo::linkmtu, MAX_REACHABLE_TIME, nd_ifinfo::maxmtu, nd6_cache_lladdr(), ND6_IFF_ACCEPT_RTADV, nd6_option_init(), nd6_options(), nd6log, ND_COMPUTE_RUNTIME, ND_IFINFO, pfxlist_onlink_check(), prelist_update(), nd_ifinfo::reachable, nd_ifinfo::recalctm, and nd_ifinfo::retrans.

Referenced by icmp6_input().

Here is the call graph for this function:



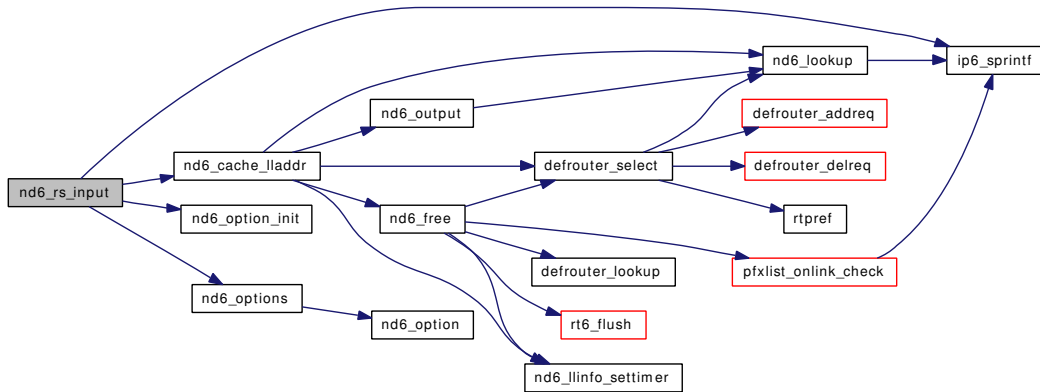
7.58.2.28 void nd6_rs_input (struct mbuf * m, int off, int icmp6len)

Definition at line 117 of file nd6_rtr.c.

References icmp6stat, IN6_IS_ADDR_UNSPECIFIED, INET6_ADDRSTRLEN, ip6_accept_rtadv, ip6_forwarding, ip6_sprintf(), nd6_cache_lladdr(), nd6_option_init(), nd6_options(), and nd6log.

Referenced by icmp6_input().

Here is the call graph for this function:



7.58.2.29 static void nd6_rtmsg(int cmd, struct rtbody *rt) [static]

Definition at line 434 of file nd6_rtr.c.

Referenced by defrouter_addreq(), defrouter_delreq(), nd6_prefix_offlink(), and nd6_prefix_onlink().

7.58.2.30 int nd6_setdefaultiface(int ifindex)

Definition at line 2098 of file nd6_rtr.c.

References scope6_setdefault().

Referenced by nd6_ioctl().

Here is the call graph for this function:



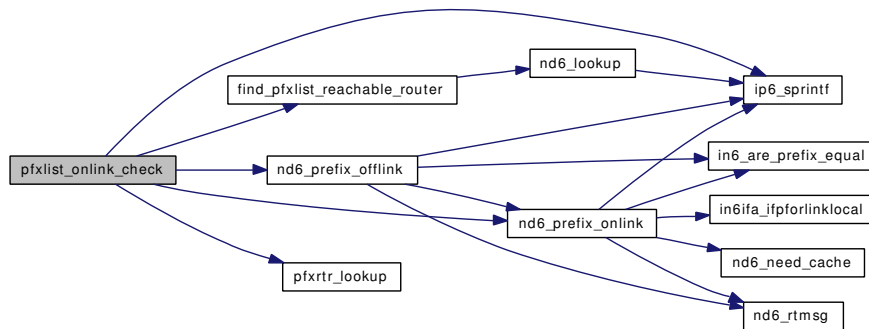
7.58.2.31 void pfxlist_onlink_check()

Definition at line 1361 of file nd6_rtr.c.

References find_pfxlist_reachable_router(), in6_ifaddr::ia6_flags, in6_ifaddr::ia6_ndpr, in6_ifaddr::ia_next, IN6_IFF_AUTOCONF, IN6_IS_ADDR_LINKLOCAL, INET6_ADDRSTRLEN, ip6_sprintf(), nd6_prefix_offlink(), nd6_prefix_onlink(), nd6log, nd_prefix::ndpr_plen, nd_prefix::ndpr_prefix, nd_prefix::ndpr_stateflags, NDPRF_DETACHED, NDPRF_ONLINK, pfxrtr_lookup(), and sockaddr_in6::sin6_addr.

Referenced by defrtrlist_del(), in6_control(), in6_unlink_ifa(), nd6_free(), nd6_na_input(), nd6_ra_input(), pfxrtr_add(), prelist_remove(), and prelist_update().

Here is the call graph for this function:



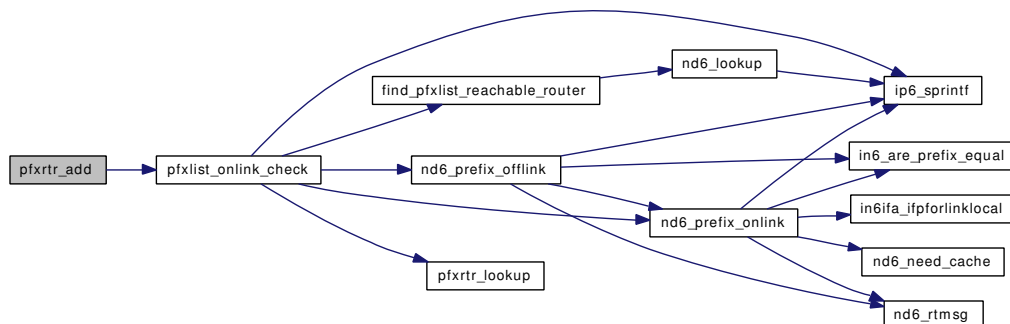
7.58.2.32 static void pfxrtr_add (struct nd_prefix * pr, struct nd_defrouter * dr) [static]

Definition at line 833 of file nd6_rtr.c.

References pfxlist_onlink_check().

Referenced by nd6_prelist_add(), and prelist_update().

Here is the call graph for this function:



7.58.2.33 static void pfxrtr_del (struct nd_pfxrouter * pfr) [static]

Definition at line 851 of file nd6_rtr.c.

Referenced by defrtrlist_del().

7.58.2.34 static struct nd_pfxrouter* pfxrtr_lookup (struct nd_prefix * pr, struct nd_defrouter * dr) [static]

Definition at line 818 of file nd6_rtr.c.

References nd_pfxrouter::router.

Referenced by defrtrlist_del(), pfxlist_onlink_check(), and prelist_update().

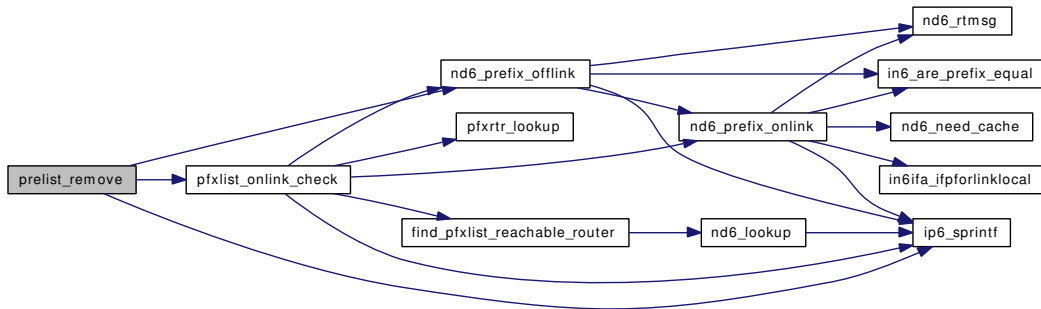
7.58.2.35 void prelist_remove (struct nd_prefix * pr)

Definition at line 938 of file nd6_rtr.c.

References INET6_ADDRSTRLEN, ip6_sprintf(), nd6_prefix_offlink(), nd6log, NDPRF_ONLINK, and pfxlist_onlink_check().

Referenced by in6_control(), nd6_ioctl(), and nd6_purge().

Here is the call graph for this function:



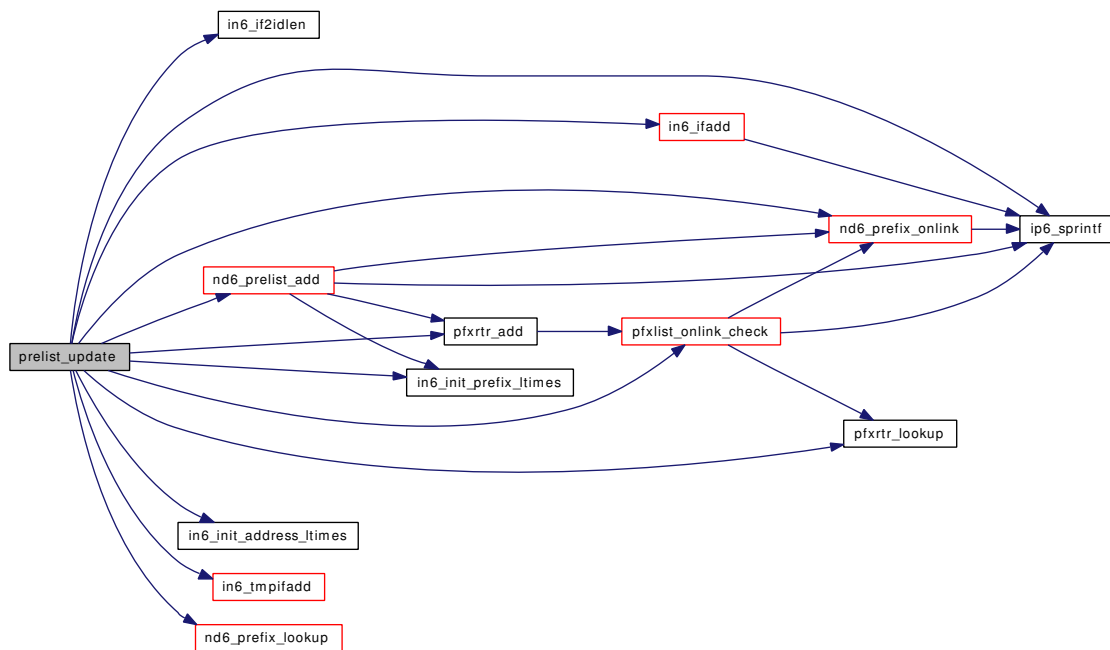
7.58.2.36 static int prelist_update (struct nd_prefixctl * new, struct nd_defrouter * dr, struct mbuf * m, int mcast) [static]

Definition at line 986 of file nd6_rtr.c.

References in6_ifaddr::ia6_createtime, in6_ifaddr::ia6_flags, in6_ifaddr::ia6_lifetime, in6_ifaddr::ia6_ndpr, in6_ifaddr::ia6_updatetime, in6_addrlifetime::ia6t_pltime, in6_addrlifetime::ia6t_vltime, in6_if2idlen(), in6_ifadd(), IN6_IFF_ANYCAST, IN6_IFF_AUTOCONF, IN6_IFF_TEMPORARY, in6_init_address_ltimes(), in6_init_prefix_ltimes(), in6_tmpifadd(), INET6_ADDRSTRLEN, ip6_sprintf(), M_AUTHIPDGM, M_AUTHIPHDR, ND6_INFINITE_LIFETIME, nd6_prefix_lookup(), nd6_prefix_onlink(), nd6_prelist_add(), nd6log, nd_prefix::ndpr_ifp, nd_prefix::ndpr_lastupdate, nd_prefix::ndpr_plen, nd_prefix::ndpr_pltime, nd_prefix::ndpr_prefix, nd_prefix::ndpr_refcnt, nd_prefix::ndpr_stateflags, nd_prefix::ndpr_vltime, NDPRF_ONLINK, pfxlist_onlink_check(), pfxrtr_add(), pfxrtr_lookup(), sockaddr_in6::sin6_addr, and TWOHOUR.

Referenced by nd6_ra_input().

Here is the call graph for this function:



7.58.2.37 static int rt6_deleteroute (struct radix_node * rn, void * arg) [static]

Definition at line 2062 of file nd6_rtr.c.

References IN6_ARE_ADDR_EQUAL, and SIN6.

Referenced by rt6_flush().

7.58.2.38 void rt6_flush (struct in6_addr * gateway, struct ifnet * ifp)

Definition at line 2042 of file nd6_rtr.c.

References IN6_IS_ADDR_LINKLOCAL, and rt6_deleteroute().

Referenced by deftrlist_del(), nd6_free(), and nd6_na_input().

Here is the call graph for this function:



7.58.2.39 static int rtpref (struct nd_defrouter * dr) [static]

Definition at line 709 of file nd6_rtr.c.

References nd_defrouter::flags, RTPREF_HIGH, RTPREF_INVALID, RTPREF_LOW, and RTPREF_MEDIUM.

Referenced by defrouter_select(), and deftrlist_update().

7.58.3 Variable Documentation

7.58.3.1 `int ip6_desync_factor`

Definition at line 92 of file `nd6_rtr.c`.

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, and `sysctl_ip6_temppltime()`.

7.58.3.2 `u_int32_t ip6_temp_preferred_lifetime = DEF_TEMP_PREFERRED_LIFETIME`

Definition at line 93 of file `nd6_rtr.c`.

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, `sysctl_ip6_temppltime()`, and `sysctl_ip6_templtime()`.

7.58.3.3 `int ip6_temp_regen_advance = TEMPADDR_REGEN_ADVANCE`

Definition at line 100 of file `nd6_rtr.c`.

Referenced by `in6_tmpaddrtimer()`, `ip6_init2()`, and `sysctl_ip6_temppltime()`.

7.58.3.4 `u_int32_t ip6_temp_valid_lifetime = DEF_TEMP_VALID_LIFETIME`

Definition at line 94 of file `nd6_rtr.c`.

Referenced by `sysctl_ip6_templtime()`.

7.58.3.5 `int ip6_use_tempaddr = 0`

Definition at line 90 of file `nd6_rtr.c`.

Referenced by `in6_control()`, and `nd6_timer()`.

7.58.3.6 `int nd6_defifindex`

Definition at line 88 of file `nd6_rtr.c`.

Referenced by `nd6_ioctl()`, and `nd6_is_new_addr_neighbor()`.

7.58.3.7 `struct ifnet* nd6_defifp` `[static]`

Definition at line 87 of file `nd6_rtr.c`.

7.58.3.8 `int nd6_recalc_reachtm_interval`

Definition at line 106 of file `nd6.c`.

Referenced by `nd6_slowtimo()`.

7.59 /usr/src/sys/netinet6/pim6.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [pim](#)

Defines

- #define [PIM_VERSION](#) 2
- #define [PIM_MINLEN](#) 8
- #define [PIM6_REG_MINLEN](#) (PIM_MINLEN+40)
- #define [PIM_REGISTER](#) 1
- #define [PIM_NULL_REGISTER](#) 0x40000000

7.59.1 Define Documentation

7.59.1.1 #define PIM6_REG_MINLEN (PIM_MINLEN+40)

Definition at line 61 of file pim6.h.

Referenced by pim6_input().

7.59.1.2 #define PIM_MINLEN 8

Definition at line 60 of file pim6.h.

Referenced by pim6_input().

7.59.1.3 #define PIM_NULL_REGISTER 0x40000000

Definition at line 69 of file pim6.h.

Referenced by pim6_input().

7.59.1.4 #define PIM_REGISTER 1

Definition at line 66 of file pim6.h.

Referenced by pim6_input().

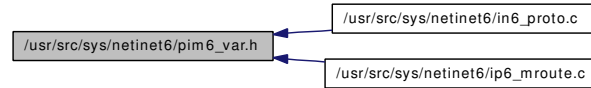
7.59.1.5 #define PIM_VERSION 2

Definition at line 43 of file pim6.h.

Referenced by pim6_input().

7.60 /usr/src/sys/netinet6/pim6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [pim6stat](#)

Defines

- #define [PIM6CTL_STATS](#) 1
- #define [PIM6CTL_MAXID](#) 2
- #define [PIM6CTL_NAMES](#)

Functions

- int [pim6_input](#) __P((struct mbuf **, int *, int))

7.60.1 Define Documentation

7.60.1.1 #define PIM6CTL_MAXID 2

Definition at line 62 of file pim6_var.h.

7.60.1.2 #define PIM6CTL_NAMES

Value:

```
{ \
    { 0, 0 }, \
    { 0, 0 }, \
}
```

Definition at line 64 of file pim6_var.h.

7.60.1.3 #define PIM6CTL_STATS 1

Definition at line 61 of file pim6_var.h.

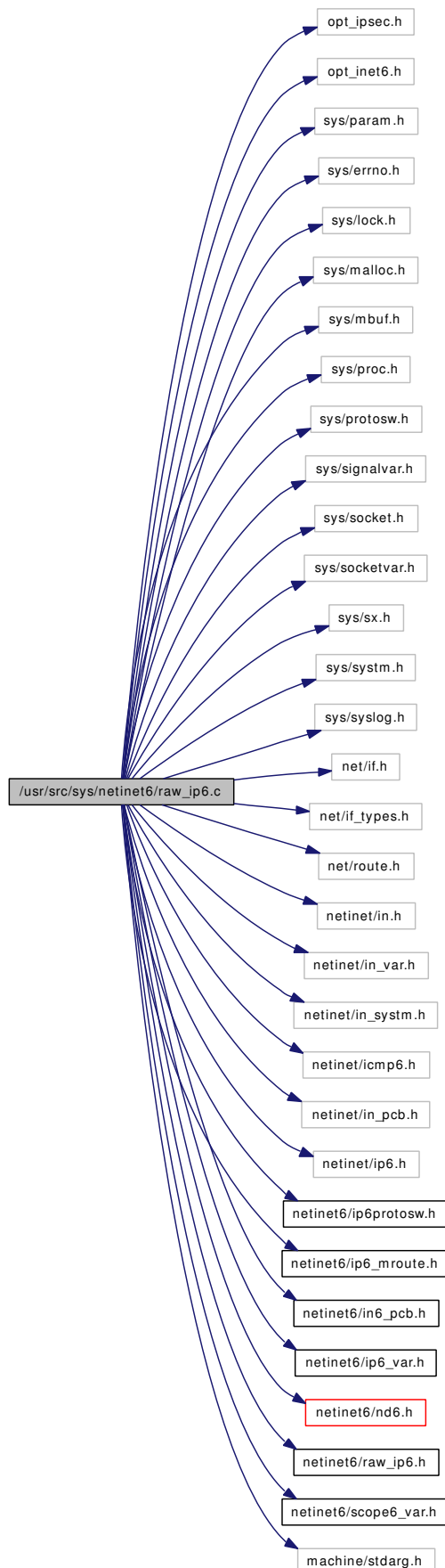
7.60.2 Function Documentation

7.60.2.1 int pim6_input __P((struct mbuf **, int *, int))

7.61 /usr/src/sys/netinet6/raw_ip6.c File Reference

```
#include "opt_ipsec.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/errno.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/proc.h>
#include <sys/protosw.h>
#include <sys/signalvar.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sx.h>
#include <sys/system.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/if_types.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/in_system.h>
#include <netinet/icmp6.h>
#include <netinet/in_pcb.h>
#include <netinet/ip6.h>
#include <netinet6/ip6protosw.h>
#include <netinet6/ip6_mroute.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/ip6_var.h>
#include <netinet6/nd6.h>
#include <netinet6/raw_ip6.h>
#include <netinet6/scope6_var.h>
#include <machine/stdarg.h>
```

Include dependency graph for raw_ip6.c:



Defines

- #define `satosin6(sa)` ((struct `sockaddr_in6` *)`(sa)`)
- #define `ifatoia6(ifa)` ((struct `in6_ifaddr` *)`(ifa)`)

Functions

- int `rip6_input` (struct mbuf **`mp`, int *`offp`, int `proto`)
- void `rip6_ctlinput` (int `cmd`, struct `sockaddr` *`sa`, void *`d`)
- int `rip6_output` (struct mbuf *`m`, va_alist)
- int `rip6_ctloutput` (struct socket *`so`, struct `sockopt` *`sopt`)
- static int `rip6_attach` (struct socket *`so`, int `proto`, struct thread *`td`)
- static void `rip6_detach` (struct socket *`so`)
- static void `rip6_abort` (struct socket *`so`)
- static void `rip6_close` (struct socket *`so`)
- static int `rip6_disconnect` (struct socket *`so`)
- static int `rip6_bind` (struct socket *`so`, struct `sockaddr` *`nam`, struct thread *`td`)
- static int `rip6_connect` (struct socket *`so`, struct `sockaddr` *`nam`, struct thread *`td`)
- static int `rip6_shutdown` (struct socket *`so`)
- static int `rip6_send` (struct socket *`so`, int `flags`, struct mbuf *`m`, struct `sockaddr` *`nam`, struct mbuf *`control`, struct thread *`td`)

Variables

- inpcbhead `ripcb`
- inpcbinfo `ripcbinfo`
- u_long `rip_sendspace`
- u_long `rip_recvspace`
- `rip6stat` `rip6stat`
- pr_usrreqs `rip6_usrreqs`

7.61.1 Define Documentation

7.61.1.1 #define `ifatoia6(ifa)` ((struct `in6_ifaddr` *)`(ifa)`)

Definition at line 111 of file `raw_ip6.c`.

7.61.1.2 #define `satosin6(sa)` ((struct `sockaddr_in6` *)`(sa)`)

Definition at line 110 of file `raw_ip6.c`.

7.61.2 Function Documentation

7.61.2.1 static void `rip6_abort` (struct socket * `so`) [static]

Definition at line 606 of file `raw_ip6.c`.

Referenced by `rip6_disconnect()`.

7.61.2.2 `static int rip6_attach (struct socket * so, int proto, struct thread * td)` [static]

Definition at line 546 of file raw_ip6.c.

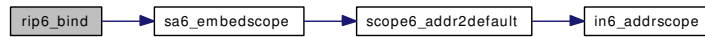
References `rip_recvspace`, `rip_sendspace`, and `ripcbinfo`.

7.61.2.3 `static int rip6_bind (struct socket * so, struct sockaddr * nam, struct thread * td)` [static]

Definition at line 640 of file raw_ip6.c.

References `IN6_IFF_ANYCAST`, `IN6_IFF_DEPRECATED`, `IN6_IFF_DETACHED`, `IN6_IFF_NOTREADY`, `IN6_IS_ADDR_UNSPECIFIED`, `ip6_use_defzone`, `ripcbinfo`, and `sa6_embedscope()`.

Here is the call graph for this function:



7.61.2.4 `static void rip6_close (struct socket * so)` [static]

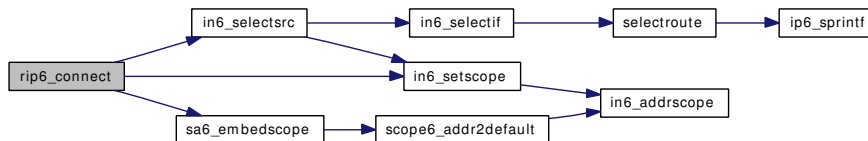
Definition at line 617 of file raw_ip6.c.

7.61.2.5 `static int rip6_connect (struct socket * so, struct sockaddr * nam, struct thread * td)` [static]

Definition at line 673 of file raw_ip6.c.

References `in6_selectsrc()`, `in6_setscope()`, `ip6_use_defzone`, `ripcbinfo`, and `sa6_embedscope()`.

Here is the call graph for this function:

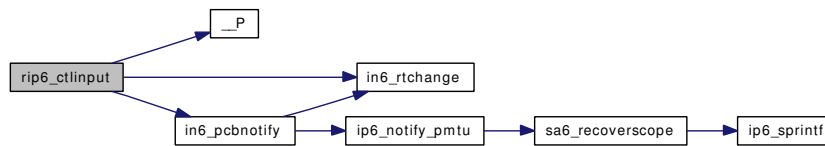


7.61.2.6 `void rip6_ctlinput (int cmd, struct sockaddr * sa, void * d)`

Definition at line 260 of file raw_ip6.c.

References `__P()`, `in6_pcbnotify()`, `in6_rtchange()`, `inet6ctlerrmap`, `ip6ctlparam::ip6c_cmdarg`, `ip6ctlparam::ip6c_ip6`, `ip6ctlparam::ip6c_m`, `ip6ctlparam::ip6c_off`, `ip6ctlparam::ip6c_src`, `ripcbinfo`, and `sa6_any`.

Here is the call graph for this function:

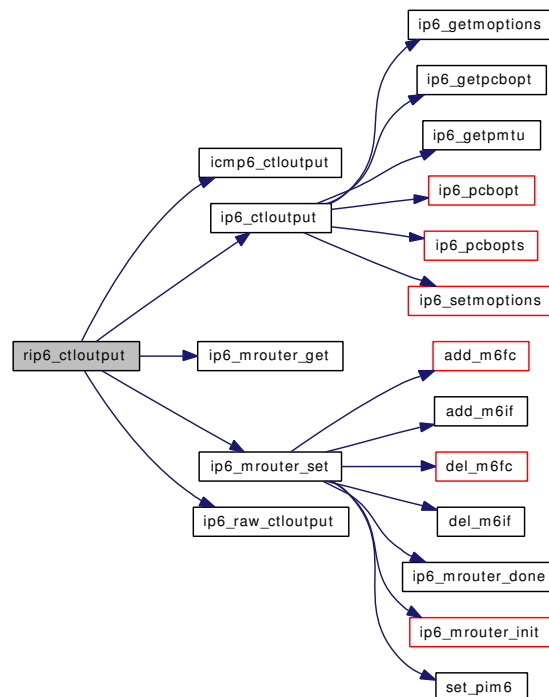


7.61.2.7 int rip6_ctloutput (struct socket * so, struct sockopt * sopt)

Definition at line 483 of file raw_ip6.c.

References icmp6_ctloutput(), ip6_ctloutput(), ip6_mrouter_get(), ip6_mrouter_set(), ip6_raw_ctloutput(), IPV6_CHECKSUM, MRT6_ADD_MFC, MRT6_ADD_MIF, MRT6_DEL_MFC, MRT6_DEL_MIF, MRT6_DONE, MRT6_INIT, and MRT6_PIM.

Here is the call graph for this function:

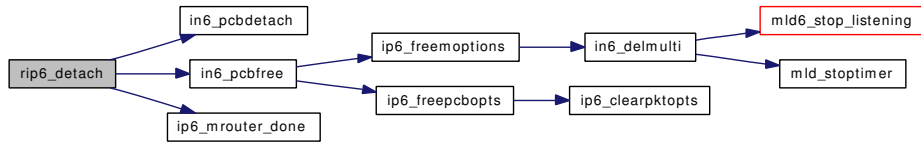


7.61.2.8 static void rip6_detach (struct socket * so) [static]

Definition at line 583 of file raw_ip6.c.

References in6_pcbdetach(), in6_pcbfree(), ip6_mrouter, ip6_mrouter_done(), and ripcbinfo.

Here is the call graph for this function:



7.61.2.9 static int rip6_disconnect (struct socket * so) [static]

Definition at line 628 of file raw_ip6.c.

References in6addr_any, and rip6_abort().

Here is the call graph for this function:



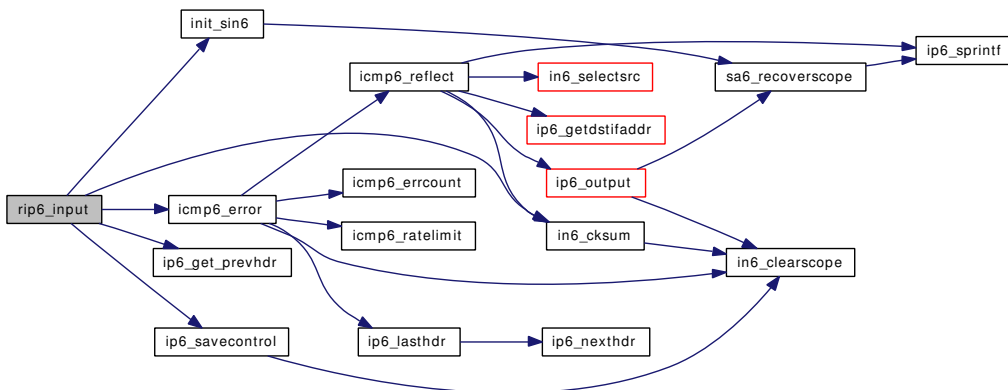
7.61.2.10 int rip6_input (struct mbuf ** mp, int * offp, int proto)

Definition at line 130 of file raw_ip6.c.

References faithprefix_p, icmp6_error(), IN6_ARE_ADDR_EQUAL, in6_cksum(), IN6_IS_ADDR_UNSPECIFIED, ipsecstat::in_polvio, init_sin6(), ip6_get_prevhdr(), ip6_savecontrol(), ipsec6stat, rip6stat::rip6s_badsum, rip6stat::rip6s_fullsock, rip6stat::rip6s_ipackets, rip6stat::rip6s_isum, rip6stat::rip6s_nosock, rip6stat::rip6s_nosockmcast, rip6stat, ripcb, and ripcbinfo.

Referenced by pim6_input().

Here is the call graph for this function:



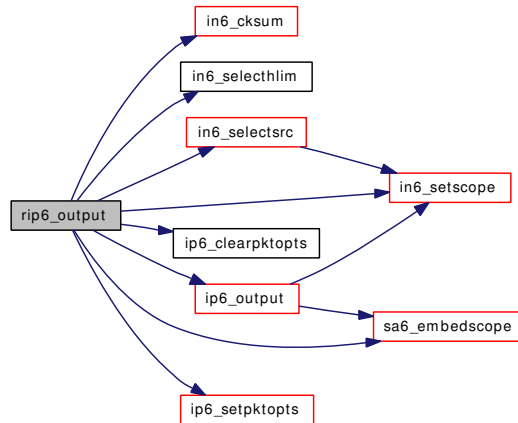
7.61.2.11 int rip6_output (struct mbuf * m, va_alist)

Definition at line 314 of file raw_ip6.c.

References icmp6stat, in6_cksum(), in6_selectlim(), in6_selectsrc(), in6_setscope(), ip6_clearpktopts(), ip6_output(), ip6_setpktopts(), ip6_use_defzone, rip6stat::rip6s_opackets, rip6stat, sa6_embedscope(), sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_scope_id.

Referenced by rip6_send().

Here is the call graph for this function:

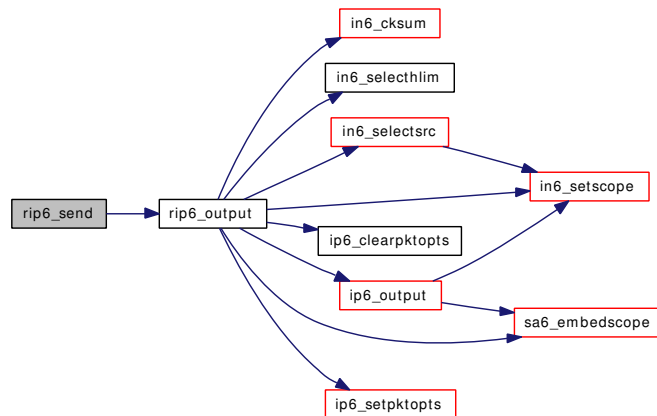


7.61.2.12 `static int rip6_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * nam, struct mbuf * control, struct thread * td) [static]`

Definition at line 743 of file `raw_ip6.c`.

References `rip6_output()`, `rip6binfo`, and `sockaddr_in6::sin6_family`.

Here is the call graph for this function:



7.61.2.13 `static int rip6_shutdown (struct socket * so) [static]`

Definition at line 730 of file `raw_ip6.c`.

7.61.3 Variable Documentation

7.61.3.1 struct pr_usrreqs [rip6_usrreqs](#)

Initial value:

```
{
    .pru_abort =          rip6_abort,
    .pru_attach =        rip6_attach,
    .pru_bind =          rip6_bind,
    .pru_connect =       rip6_connect,
    .pru_control =       in6_control,
    .pru_detach =        rip6_detach,
    .pru_disconnect =    rip6_disconnect,
    .pru_peeraddr =      in6_setpeeraddr,
    .pru_send =          rip6_send,
    .pru_shutdown =      rip6_shutdown,
    .pru_sockaddr =      in6_setsockaddr,
    .pru_close =         rip6_close,
}
```

Definition at line 802 of file raw_ip6.c.

7.61.3.2 struct [rip6stat](#) [rip6stat](#)

Definition at line 122 of file raw_ip6.c.

Referenced by [rip6_input\(\)](#), and [rip6_output\(\)](#).

7.61.3.3 u_long [rip_recvspace](#)

Referenced by [rip6_attach\(\)](#).

7.61.3.4 u_long [rip_sendspace](#)

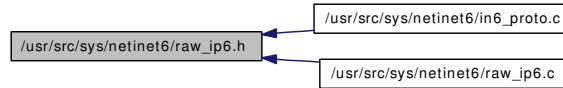
Referenced by [rip6_attach\(\)](#).

7.61.3.5 struct inpcbhead [ripcb](#)

7.61.3.6 struct inpcbinfo [ripcbinfo](#)

7.62 /usr/src/sys/netinet6/raw_ip6.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [rip6stat](#)

Variables

- [rip6stat](#) `rip6stat`

7.62.1 Variable Documentation

7.62.1.1 struct [rip6stat](#) `rip6stat`

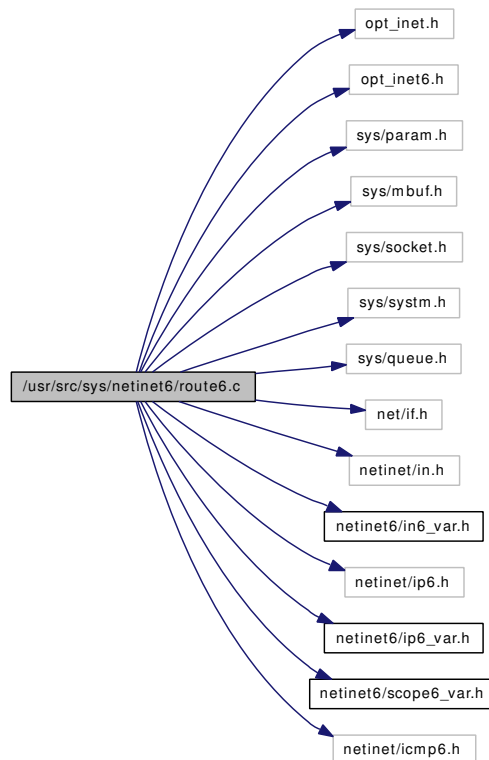
Definition at line 122 of file `raw_ip6.c`.

Referenced by `rip6_input()`, and `rip6_output()`.

7.63 /usr/src/sys/netinet6/route6.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include <sys/param.h>  
#include <sys/mbuf.h>  
#include <sys/socket.h>  
#include <sys/system.h>  
#include <sys/queue.h>  
#include <net/if.h>  
#include <netinet/in.h>  
#include <netinet6/in6_var.h>  
#include <netinet/ip6.h>  
#include <netinet6/ip6_var.h>  
#include <netinet6/scope6_var.h>  
#include <netinet/icmp6.h>
```

Include dependency graph for route6.c:



Functions

- static int `ip6_rthdr0 __P` ((struct mbuf *, struct ip6_hdr *, struct ip6_rthdr0 *))
- int `route6_input` (struct mbuf **mp, int *offp, int proto)
- static int `ip6_rthdr0` (struct mbuf *m, struct ip6_hdr *ip6, struct ip6_rthdr0 *rh0)

7.63.1 Function Documentation

7.63.1.1 static int ip6_rthdr0 __P ((struct mbuf *, struct ip6_hdr *, struct ip6_rthdr0 *))

[static]

7.63.1.2 static int ip6_rthdr0 (struct mbuf * m, struct ip6_hdr * ip6, struct ip6_rthdr0 * rh0)

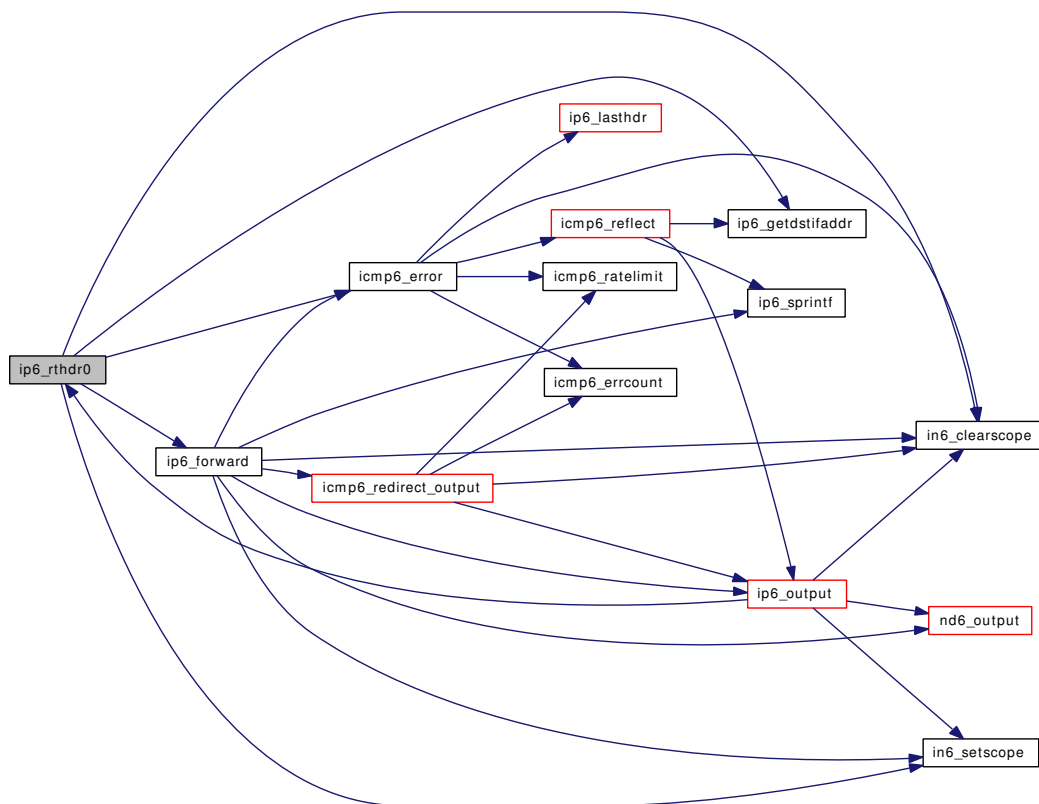
[static]

Definition at line 140 of file route6.c.

References `icmp6_error()`, `in6_clearscope()`, `IN6_IS_ADDR_MULTICAST`, `IN6_IS_ADDR_UNSPECIFIED`, `IN6_IS_ADDR_V4COMPAT`, `IN6_IS_ADDR_V4MAPPED`, `in6_setscope()`, `ip6_forward()`, and `ip6_getdstifaddr()`.

Referenced by `icmp6_notify_error()`, `ip6_output()`, and `route6_input()`.

Here is the call graph for this function:

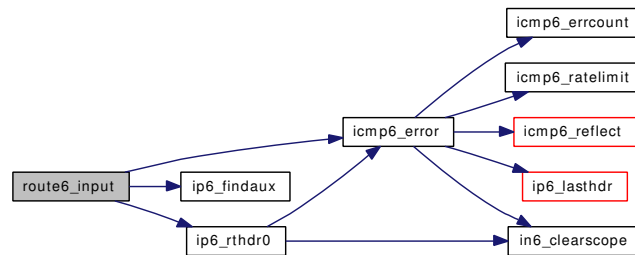


7.63.1.3 int route6_input (struct mbuf ** mp, int * offp, int proto)

Definition at line 56 of file route6.c.

References icmp6_error(), ip6_findaux(), ip6_rthdr0(), ip6aux::ip6a_flags, IP6A_SWAP, and IPV6_RTHDR_TYPE_0.

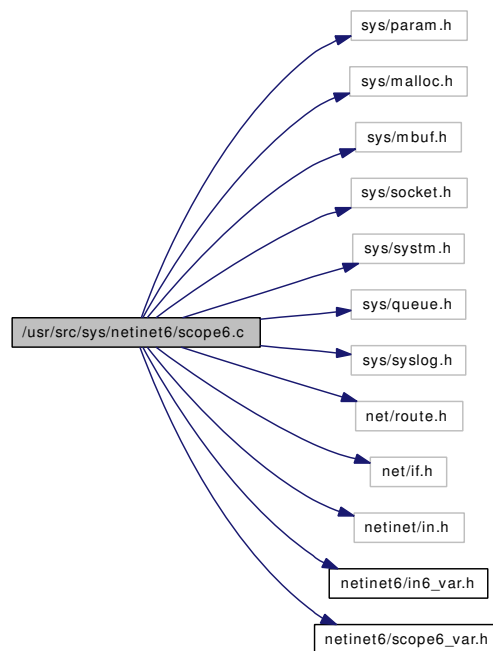
Here is the call graph for this function:



7.64 /usr/src/sys/netinet6/scope6.c File Reference

```
#include <sys/param.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/system.h>
#include <sys/queue.h>
#include <sys/syslog.h>
#include <net/route.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet6/in6_var.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for scope6.c:



Defines

- #define [SCOPE6_LOCK_INIT\(\)](#) `mtx_init(&scope6_lock, "scope6_lock", NULL, MTX_DEF)`
- #define [SCOPE6_LOCK\(\)](#) `mtx_lock(&scope6_lock)`
- #define [SCOPE6_UNLOCK\(\)](#) `mtx_unlock(&scope6_lock)`
- #define [SCOPE6_LOCK_ASSERT\(\)](#) `mtx_assert(&scope6_lock, MA_OWNED)`
- #define [SID\(ifp\)](#) `((struct in6_ifextra *) (ifp)) → if_afdata[AF_INET6] → scope6_id`

Functions

- void `scope6_init` ()
- `scope6_id * scope6_ifattach` (struct ifnet *ifp)
- void `scope6_ifdetach` (struct `scope6_id` *sid)
- int `scope6_set` (struct ifnet *ifp, struct `scope6_id` *idlist)
- int `scope6_get` (struct ifnet *ifp, struct `scope6_id` *idlist)
- int `in6_addrscope` (struct `in6_addr` *addr)
- void `scope6_setdefault` (struct ifnet *ifp)
- int `scope6_get_default` (struct `scope6_id` *idlist)
- u_int32_t `scope6_addr2default` (struct `in6_addr` *addr)
- int `sa6_embedscope` (struct `sockaddr_in6` *sin6, int defaulttok)
- int `sa6_recoverscope` (struct `sockaddr_in6` *sin6)
- int `in6_setscope` (struct `in6_addr` *in6, struct ifnet *ifp, u_int32_t *ret_id)
- int `in6_clearscope` (struct `in6_addr` *in6)

Variables

- int `ip6_use_defzone` = 0
- static struct mtx `scope6_lock`
- static struct `scope6_id` `sid_default`

7.64.1 Define Documentation

7.64.1.1 #define SCOPE6_LOCK() mtx_lock(&scope6_lock)

Definition at line 61 of file `scope6.c`.

Referenced by `in6_setscope()`, `scope6_addr2default()`, `scope6_get()`, `scope6_get_default()`, `scope6_set()`, and `scope6_setdefault()`.

7.64.1.2 #define SCOPE6_LOCK_ASSERT() mtx_assert(&scope6_lock, MA_OWNED)

Definition at line 63 of file `scope6.c`.

7.64.1.3 #define SCOPE6_LOCK_INIT() mtx_init(&scope6_lock, "scope6_lock", NULL, MTX_DEF)

Definition at line 60 of file `scope6.c`.

Referenced by `scope6_init()`.

7.64.1.4 #define SCOPE6_UNLOCK() mtx_unlock(&scope6_lock)

Definition at line 62 of file `scope6.c`.

Referenced by `in6_setscope()`, `scope6_addr2default()`, `scope6_get()`, `scope6_get_default()`, `scope6_set()`, and `scope6_setdefault()`.

7.64.1.5 #define SID(ifu) (((struct in6_ifextra *))(ifu) → if_afdata[AF_INET6]) → scope6_id)

Definition at line 66 of file scope6.c.

Referenced by in6_setscope(), scope6_get(), and scope6_set().

7.64.2 Function Documentation**7.64.2.1 int in6_addrscope (struct in6_addr * addr)**

Definition at line 205 of file scope6.c.

References in6addr_loopback, IPV6_ADDR_SCOPE_GLOBAL, IPV6_ADDR_SCOPE_INTFACELOCAL, IPV6_ADDR_SCOPE_LINKLOCAL, and IPV6_ADDR_SCOPE_SITELOCAL.

Referenced by in6_ifawithifu(), in6_setscope(), ni6_addrs(), ni6_store_addrs(), and scope6_addr2default().

7.64.2.2 int in6_clearscope (struct in6_addr * in6)

Definition at line 484 of file scope6.c.

References IN6_IS_ADDR_MC_INTFACELOCAL, and IN6_IS_SCOPE_LINKLOCAL.

Referenced by icmp6_error(), icmp6_redirect_output(), in6_cksum(), in6_lifaddr_ioctl(), ip6_forward(), ip6_input(), ip6_mloopback(), ip6_output(), ip6_rthdr0(), ip6_savecontrol(), mld6_sendpkt(), nd6_ioctl(), nd6_na_output(), nd6_ns_output(), and ni6_store_addrs().

7.64.2.3 int in6_setscope (struct in6_addr * in6, struct ifnet * ifp, u_int32_t * ret_id)

Definition at line 407 of file scope6.c.

References in6_addrscope(), IN6_IS_ADDR_LOOPBACK, IN6_IS_ADDR_MC_INTFACELOCAL, IN6_IS_SCOPE_LINKLOCAL, IPV6_ADDR_SCOPE_INTFACELOCAL, IPV6_ADDR_SCOPE_LINKLOCAL, IPV6_ADDR_SCOPE_ORGLOCAL, IPV6_ADDR_SCOPE_SITELOCAL, scope6_id::s6id_list, SCOPE6_LOCK, SCOPE6_UNLOCK, and SID.

Referenced by icmp6_error2(), icmp6_mtudisc_update(), icmp6_notify_error(), icmp6_redirect_input(), in6_control(), in6_ifattach_linklocal(), in6_ifdetach(), in6_nigroup(), in6_pcbladdr(), in6_purgeaddr(), in6_selectsrc(), in6_update_ifa(), ip6_forward(), ip6_input(), ip6_mdq(), ip6_output(), ip6_rthdr0(), ip6_setmoptions(), mld6_input(), mld6_start_listening(), mld6_stop_listening(), nd6_ioctl(), nd6_is_new_addr_neighbor(), nd6_na_input(), nd6_na_output(), nd6_ns_input(), nd6_ns_output(), ni6_input(), rip6_connect(), rip6_output(), and udp6_output().

Here is the call graph for this function:

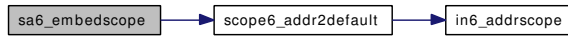
**7.64.2.4 int sa6_embedscope (struct sockaddr_in6 * sin6, int defaultok)**

Definition at line 331 of file scope6.c.

References IN6_IS_ADDR_MC_INTFACELOCAL, IN6_IS_SCOPE_LINKLOCAL, and scope6_addr2default().

Referenced by in6_control(), in6_pcbbind(), in6_pcblladdr(), in6_update_ifa(), ip6_output(), ip6_setmoptions(), ip6_setpktopt(), rip6_bind(), rip6_connect(), rip6_output(), udp6_getcred(), and udp6_output().

Here is the call graph for this function:



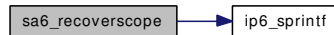
7.64.2.5 int sa6_recoverscope (struct sockaddr_in6 * sin6)

Definition at line 369 of file scope6.c.

References IN6_IS_ADDR_MC_INTFACELOCAL, IN6_IS_SCOPE_LINKLOCAL, INET6_ADDRSTRLEN, and ip6_sprintf().

Referenced by icmp6_rip6_input(), in6_control(), in6_lifaddr_ioctl(), in6_sockaddr(), init_sin6(), ip6_notify_pmtu(), ip6_output(), nd6_is_new_addr_neighbor(), nd6_sysctl_drlist(), nd6_sysctl_prlist(), sctp6_getaddr(), and sctp6_peeraddr().

Here is the call graph for this function:



7.64.2.6 u_int32_t scope6_addr2default (struct in6_addr * addr)

Definition at line 300 of file scope6.c.

References in6_addrscope(), IN6_IS_ADDR_LOOPBACK, scope6_id::s6id_list, SCOPE6_LOCK, SCOPE6_UNLOCK, and sid_default.

Referenced by sa6_embedscope().

Here is the call graph for this function:



7.64.2.7 int scope6_get (struct ifnet * ifp, struct scope6_id * idlist)

Definition at line 179 of file scope6.c.

References SCOPE6_LOCK, SCOPE6_UNLOCK, and SID.

Referenced by in6_control().

7.64.2.8 int scope6_get_default (struct scope6_id * idlist)

Definition at line 288 of file scope6.c.

References SCOPE6_LOCK, SCOPE6_UNLOCK, and sid_default.

Referenced by in6_control().

7.64.2.9 struct [scope6_id](#)* scope6_ifattach (struct ifnet * ifp)

Definition at line 78 of file scope6.c.

References IPV6_ADDR_SCOPE_INTFACELOCAL, IPV6_ADDR_SCOPE_LINKLOCAL, IPV6_ADDR_SCOPE_ORGLOCAL, and IPV6_ADDR_SCOPE_SITELOCAL.

Referenced by in6_domifattach().

7.64.2.10 void scope6_ifdetach (struct [scope6_id](#) * sid)

Definition at line 102 of file scope6.c.

Referenced by in6_domifdetach().

7.64.2.11 void scope6_init ()

Definition at line 70 of file scope6.c.

References SCOPE6_LOCK_INIT, and sid_default.

7.64.2.12 int scope6_set (struct ifnet * ifp, struct [scope6_id](#) * idlist)

Definition at line 110 of file scope6.c.

References IPV6_ADDR_SCOPE_INTFACELOCAL, IPV6_ADDR_SCOPE_LINKLOCAL, scope6_id::s6id_list, SCOPE6_LOCK, SCOPE6_UNLOCK, and SID.

Referenced by in6_control().

7.64.2.13 void scope6_setdefault (struct ifnet * ifp)

Definition at line 265 of file scope6.c.

References IPV6_ADDR_SCOPE_INTFACELOCAL, IPV6_ADDR_SCOPE_LINKLOCAL, scope6_id::s6id_list, SCOPE6_LOCK, SCOPE6_UNLOCK, and sid_default.

Referenced by nd6_setdefaultiface().

7.64.3 Variable Documentation

7.64.3.1 int [ip6_use_defzone](#) = 0

Definition at line 52 of file scope6.c.

Referenced by in6_pcbbind(), in6_pcblladdr(), ip6_output(), ip6_setmoptions(), ip6_setpktopt(), rip6_bind(), rip6_connect(), rip6_output(), udp6_getcred(), and udp6_output().

7.64.3.2 struct mtx [scope6_lock](#) [static]

Definition at line 59 of file scope6.c.

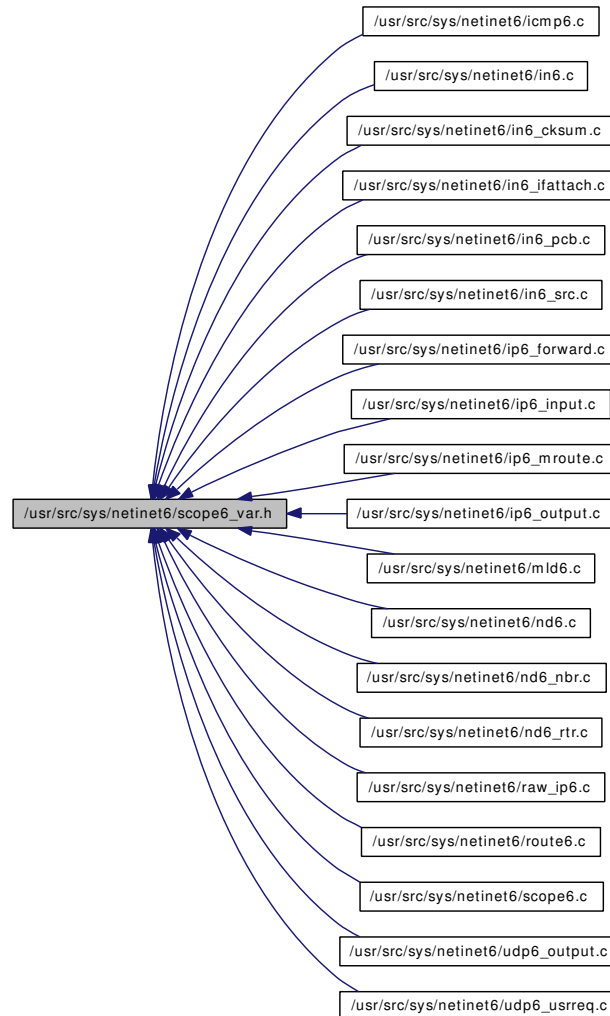
7.64.3.3 struct [scope6_id sid_default](#) [static]

Definition at line 65 of file scope6.c.

Referenced by [scope6_addr2default\(\)](#), [scope6_get_default\(\)](#), [scope6_init\(\)](#), and [scope6_setdefault\(\)](#).

7.65 /usr/src/sys/netinet6/scope6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [scope6_id](#)

Functions

- void [scope6_init](#) __P ((void))
- [scope6_id](#) *[scope6_ifattach](#) __P ((struct ifnet *))
- void [scope6_ifdetach](#) __P ((struct [scope6_id](#) *))
- int [scope6_set](#) __P ((struct ifnet *, struct [scope6_id](#) *))
- u_int32_t [scope6_in6_addrscope](#) __P ((struct [in6_addr](#) *))
- int [sa6_embedscope](#) __P ((struct [sockaddr_in6](#) *, int))
- int [sa6_recoverscope](#) __P ((struct [sockaddr_in6](#) *))
- int [in6_setscope](#) __P ((struct [in6_addr](#) *, struct ifnet *, u_int32_t *))

7.65.1 Function Documentation

7.65.1.1 `int in6_setscope __P((struct in6_addr *, struct ifnet *, u_int32_t *))`

7.65.1.2 `int sa6_recoverscope __P((struct sockaddr_in6 *))`

7.65.1.3 `int sa6_embedscope __P((struct sockaddr_in6 *, int))`

7.65.1.4 `u_int32_t scope6_in6_addrscope __P((struct in6_addr *))`

7.65.1.5 `int scope6_get __P((struct ifnet *, struct scope6_id *))`

7.65.1.6 `int scope6_get_default __P((struct scope6_id *))`

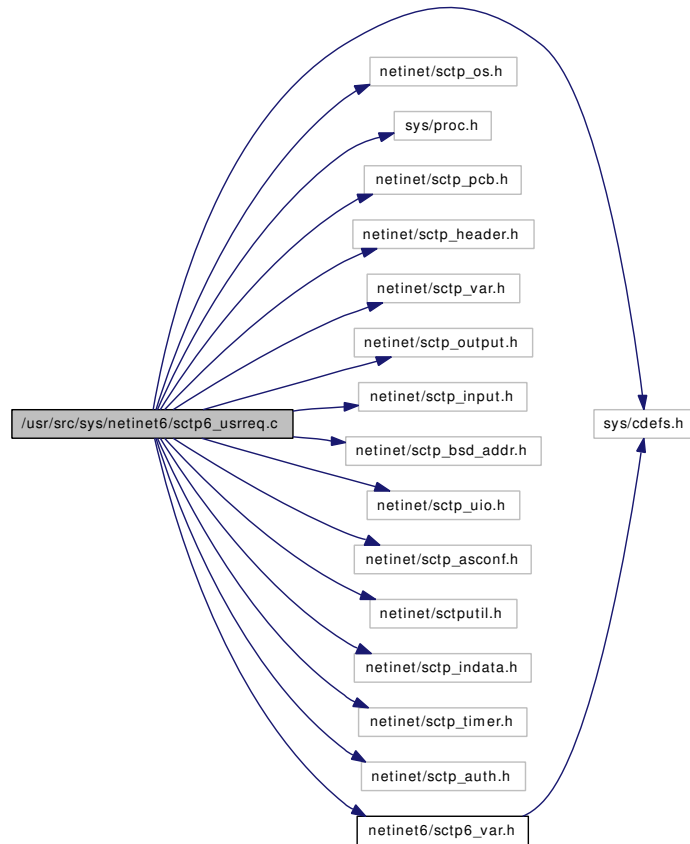
7.65.1.7 `struct scope6_id* scope6_ifattach __P((struct ifnet *))`

7.65.1.8 `void scope6_init __P((void))`

7.66 /usr/src/sys/netinet6/sctp6_usrreq.c File Reference

```
#include <sys/cdefs.h>
#include <netinet/sctp_os.h>
#include <sys/proc.h>
#include <netinet/sctp_pcb.h>
#include <netinet/sctp_header.h>
#include <netinet/sctp_var.h>
#include <netinet/sctp_output.h>
#include <netinet/sctp_input.h>
#include <netinet/sctp_bsd_addr.h>
#include <netinet/sctp_uio.h>
#include <netinet/sctp_asconf.h>
#include <netinet/sctputil.h>
#include <netinet/sctp_indata.h>
#include <netinet/sctp_timer.h>
#include <netinet/sctp_auth.h>
#include <netinet6/sctp6_var.h>
```

Include dependency graph for sctp6_usrreq.c:



Functions

- `__FBSDID` ("\$FreeBSD: src/sys/netinet6/sctp6_usrreq.c,v 1.10 2007/02/12 23:24:31 rrs Exp \$")
- `int sctp6_input` (struct mbuf **mp, int *offp, int proto)
- `static void sctp6_notify_mbuf` (struct sctp_inpcb *inp, struct icmp6_hdr *icmp6, struct sctphdr *sh, struct sctp_tcb *stcb, struct sctp_nets *net)
- `void sctp6_ctlinput` (int cmd, struct sockaddr *pkt_dst, void *d)
- `static int sctp6_getcred` (SYSCTL_HANDLER_ARGS)
- `SYSCTL_PROC` (_net_inet6_sctp6, OID_AUTO, getcred, CTLTYPE_OPAQUE|CTLFLAG_RW, 0, 0, sctp6_getcred, "S,ucrd", "Get the ucred of a SCTP6 connection")
- `static void sctp6_abort` (struct socket *so)
- `static int sctp6_attach` (struct socket *so, int proto, struct thread *p)
- `static int sctp6_bind` (struct socket *so, struct sockaddr *addr, struct thread *p)
- `static void sctp6_close` (struct socket *so)
- `static int sctp6_disconnect` (struct socket *so)
- `int sctp6_sendm` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *p)
- `static int sctp6_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *p)
- `static int sctp6_connect` (struct socket *so, struct sockaddr *addr, struct thread *p)
- `static int sctp6_getaddr` (struct socket *so, struct sockaddr **addr)
- `static int sctp6_peeraddr` (struct socket *so, struct sockaddr **addr)
- `static int sctp6_in6getaddr` (struct socket *so, struct sockaddr **nam)
- `static int sctp6_getpeeraddr` (struct socket *so, struct sockaddr **nam)

Variables

- protosw [inetsw](#) []
- int [sctp_no_csum_on_loopback](#)
- pr_usrreqs [sctp6_usrreqs](#)

7.66.1 Function Documentation

7.66.1.1 `__FBSDID("$FreeBSD: src/sys/netinet6/sctp6_usrreq.c, v 1.10 2007/02/12 23:24:31 rrs Exp $")`

7.66.1.2 `static void sctp6_abort (struct socket * so) [static]`

Definition at line 500 of file sctp6_usrreq.c.

7.66.1.3 `static int sctp6_attach (struct socket * so, int proto, struct thread * p) [static]`

Definition at line 546 of file sctp6_usrreq.c.

References [in6pcb](#).

7.66.1.4 `static int sctp6_bind (struct socket * so, struct sockaddr * addr, struct thread * p) [static]`

Definition at line 587 of file sctp6_usrreq.c.

References [IN6_IS_ADDR_UNSPECIFIED](#), [IN6_IS_ADDR_V4MAPPED](#), [in6_sin6_2_sin\(\)](#), [in6pcb](#), and [sockaddr_in6::sin6_addr](#).

Referenced by [sctp6_connect\(\)](#).

Here is the call graph for this function:



7.66.1.5 `static void sctp6_close (struct socket * so) [static]`

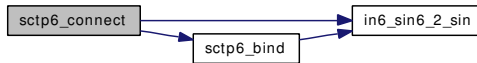
Definition at line 644 of file sctp6_usrreq.c.

7.66.1.6 `static int sctp6_connect (struct socket * so, struct sockaddr * addr, struct thread * p) [static]`

Definition at line 943 of file sctp6_usrreq.c.

References [IN6_IS_ADDR_V4MAPPED](#), [in6_sin6_2_sin\(\)](#), [in6pcb](#), [ip6_v6only](#), [sctp6_bind\(\)](#), [sin6](#), and [sockaddr_in6::sin6_addr](#).

Here is the call graph for this function:

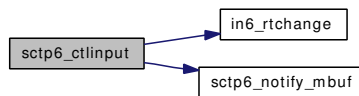


7.66.1.7 void sctp6_ctlinput (int cmd, struct sockaddr * pktdst, void * d)

Definition at line 335 of file sctp6_usrreq.c.

References in6_rtchange(), in6pcb, inet6ctlerrmap, and sctp6_notify_mbuf().

Here is the call graph for this function:



7.66.1.8 static int sctp6_disconnect (struct socket * so) [static]

Definition at line 708 of file sctp6_usrreq.c.

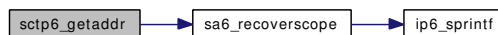
7.66.1.9 static int sctp6_getaddr (struct socket * so, struct sockaddr ** addr) [static]

Definition at line 1065 of file sctp6_usrreq.c.

References sa6_recoverscope(), sin6, sockaddr_in6::sin6_addr, and sockaddr_in6::sin6_family.

Referenced by sctp6_in6getaddr().

Here is the call graph for this function:



7.66.1.10 static int sctp6_getcred (SYSCTL_HANDLER_ARGS) [static]

Definition at line 432 of file sctp6_usrreq.c.

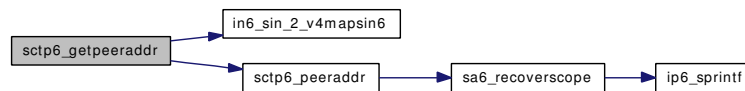
References sin6tosn.

7.66.1.11 static int sctp6_getpeeraddr (struct socket * so, struct sockaddr ** nam) [static]

Definition at line 1252 of file sctp6_usrreq.c.

References in6_sin_2_v4mapsin6(), in6pcb, sctp6_peeraddr(), and sin6.

Here is the call graph for this function:

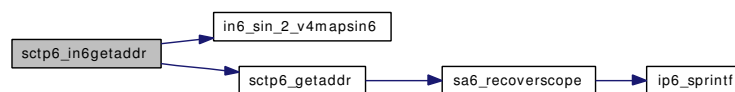


7.66.1.12 `static int sctp6_in6getaddr (struct socket * so, struct sockaddr ** nam) [static]`

Definition at line 1220 of file sctp6_usrreq.c.

References `in6_sin_2_v4mapsin6()`, `in6pcb`, `sctp6_getaddr()`, and `sin6`.

Here is the call graph for this function:



7.66.1.13 `int sctp6_input (struct mbuf ** mp, int * offp, int proto)`

Definition at line 65 of file sctp6_usrreq.c.

References `faithprefix_p`, `IN6_ARE_ADDR_EQUAL`, `IN6_IS_ADDR_MULTICAST`, `ipsecstat::in_polvio`, `ipsec6stat`, and `sctp_no_csum_on_loopback`.

7.66.1.14 `static void sctp6_notify_mbuf (struct sctp_inpcb * inp, struct icmp6_hdr * icmp6, struct sctphdr * sh, struct sctp_tcb * tcb, struct sctp_nets * net) [static]`

Definition at line 259 of file sctp6_usrreq.c.

Referenced by `sctp6_ctlinput()`.

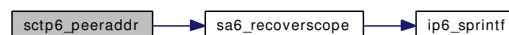
7.66.1.15 `static int sctp6_peeraddr (struct socket * so, struct sockaddr ** addr) [static]`

Definition at line 1157 of file sctp6_usrreq.c.

References `sa6_recoverscope()`, `sin6`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_family`, and `sockaddr_in6::sin6_port`.

Referenced by `sctp6_getpeeraddr()`.

Here is the call graph for this function:

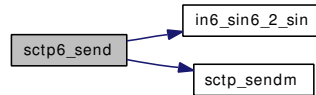


7.66.1.16 `static int sctp6_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * addr, struct mbuf * control, struct thread * p) [static]`

Definition at line 835 of file sctp6_usrreq.c.

References IN6_IS_ADDR_V4MAPPED, in6_sin6_2_sin(), in6pcb, ip6_v6only, sctp_sendm(), sin6, and sockaddr_in6::sin6_addr.

Here is the call graph for this function:



7.66.1.17 int sctp_sendm (struct socket * so, int flags, struct mbuf * m, struct sockaddr * addr, struct mbuf * control, struct thread * p)

Referenced by sctp6_send().

7.66.1.18 SYSCTL_PROC (_net_inet6_sctp6, OID_AUTO, getcred, CTLTYPE_OPAQUE|CTLFLAG_RW, 0, 0, sctp6_getcred, " S, ucred", "Get the ucred of a SCTP6 connection")

7.66.2 Variable Documentation

7.66.2.1 struct protosw inetsw[]

Referenced by udp6_abort(), udp6_close(), udp6_disconnect(), and udp6_send().

7.66.2.2 struct pr_usrreqs sctp6_usrreqs

Initial value:

```

{
    .pru_abort = sctp6_abort,
    .pru_accept = sctp_accept,
    .pru_attach = sctp6_attach,
    .pru_bind = sctp6_bind,
    .pru_connect = sctp6_connect,
    .pru_control = in6_control,
    .pru_close = sctp6_close,
    .pru_detach = sctp6_close,
    .pru_sopoll = sopoll_generic,
    .pru_disconnect = sctp6_disconnect,
    .pru_listen = sctp_listen,
    .pru_peeraddr = sctp6_getpeeraddr,
    .pru_send = sctp6_send,
    .pru_shutdown = sctp_shutdown,
    .pru_sockaddr = sctp6_in6getaddr,
    .pru_sosend = sctp_sosend,
    .pru_soreceive = sctp_soreceive
}
  
```

Definition at line 1281 of file sctp6_usrreq.c.

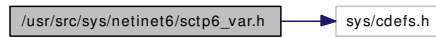
7.66.2.3 int sctp_no_csum_on_loopback

Referenced by sctp6_input().

7.67 /usr/src/sys/netinet6/sctp6_var.h File Reference

```
#include <sys/cdefs.h>
```

Include dependency graph for sctp6_var.h:



This graph shows which files directly or indirectly include this file:



Functions

- [__FBSDID](#) ("\$FreeBSD: src/sys/netinet6/sctp6_var.h,v 1.3 2007/01/18 09:58:43 rrs Exp \$")
- [SYSCTL_DECL](#) (_net_inet6_sctp6)
- [int sctp6_ctloutput](#) __P ((struct socket *, struct sockopt *))
- [int sctp6_input](#) __P ((struct mbuf **, int *, int))
- [int sctp6_output](#) __P ((struct sctp_inpcb *, struct mbuf *, struct sockaddr *, struct mbuf *, struct proc *))
- [void sctp6_ctlinput](#) __P ((int, struct sockaddr *, void *))

Variables

- [pr_usrreqs](#) [sctp6_usrreqs](#)

7.67.1 Function Documentation

7.67.1.1 [__FBSDID](#) ("\$FreeBSD: src/sys/netinet6/sctp6_var. h, v 1.3 2007/01/18 09:58:43 rrs Exp \$")

7.67.1.2 [void sctp6_ctlinput](#) __P ((int, struct sockaddr *, void *))

7.67.1.3 [int sctp6_output](#) __P ((struct sctp_inpcb *, struct mbuf *, struct sockaddr *, struct mbuf *, struct proc *))

7.67.1.4 [int sctp6_input](#) __P ((struct mbuf **, int *, int))

7.67.1.5 [int sctp6_ctloutput](#) __P ((struct socket *, struct sockopt *))

7.67.1.6 [SYSCTL_DECL](#) (_net_inet6_sctp6)

7.67.2 Variable Documentation

7.67.2.1 [struct pr_usrreqs](#) [sctp6_usrreqs](#)

Definition at line 1281 of file sctp6_usrreq.c.

7.68 /usr/src/sys/netinet6/tcp6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void tcp6_ctlinput __P ((int, struct sockaddr *, void *))
- void tcp6_init __P ((void))
- int tcp6_input __P ((struct mbuf **, int *, int))
- rentry * tcp_rtlookup6 (struct in_conninfo *)

Variables

- int tcp_v6mssdflt
- pr_usrreqs tcp6_usrreqs

7.68.1 Function Documentation

7.68.1.1 int tcp6_input __P ((struct mbuf **, int *, int))

7.68.1.2 void tcp6_init __P ((void))

7.68.1.3 void tcp6_ctlinput __P ((int, struct sockaddr *, void *))

7.68.1.4 struct rentry* tcp_rtlookup6 (struct in_conninfo *)

7.68.2 Variable Documentation

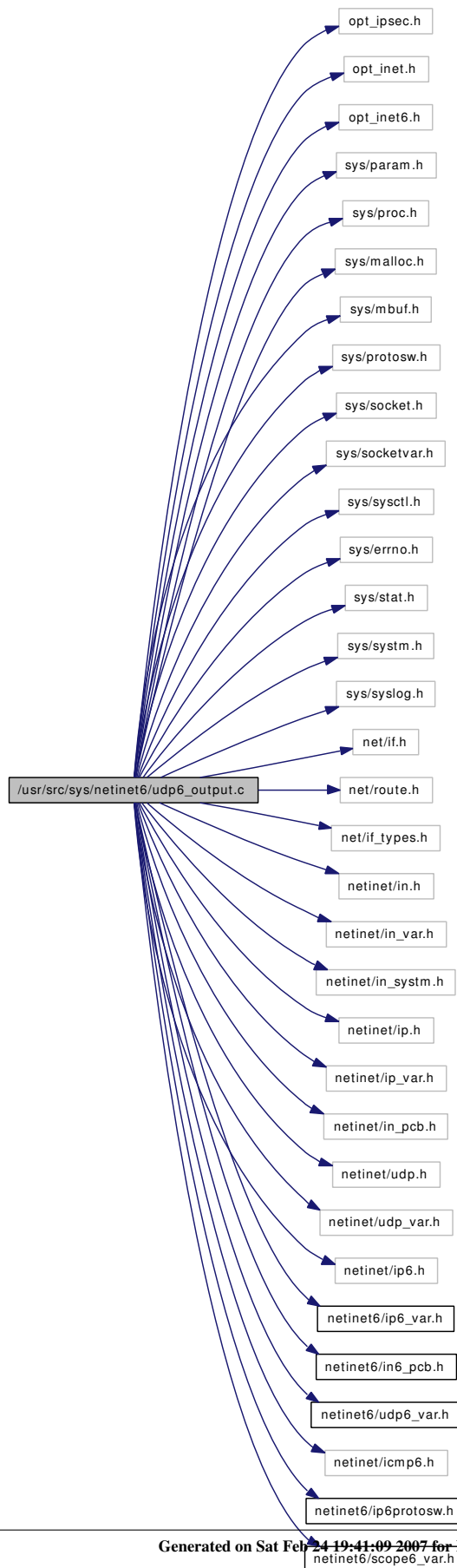
7.68.2.1 struct pr_usrreqs tcp6_usrreqs

7.68.2.2 int tcp_v6mssdflt

7.69 /usr/src/sys/netinet6/udp6_output.c File Reference

```
#include "opt_ipsec.h"
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/proc.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/errno.h>
#include <sys/stat.h>
#include <sys/system.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <net/if_types.h>
#include <netinet/in.h>
#include <netinet/in_var.h>
#include <netinet/in_system.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/in_pcb.h>
#include <netinet/udp.h>
#include <netinet/udp_var.h>
#include <netinet/ip6.h>
#include <netinet6/ip6_var.h>
#include <netinet6/in6_pcb.h>
#include <netinet6/udp6_var.h>
#include <netinet/icmp6.h>
#include <netinet6/ip6protosw.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for udp6_output.c:



Defines

- #define `in6pcb` `inpcb`
- #define `udp6stat` `udpstat`
- #define `udp6s_opackets` `udps_opackets`

Functions

- int `udp6_output` (struct `in6pcb` *`in6p`, struct `mbuf` *`m`, struct `sockaddr` *`addr6`, struct `mbuf` *`control`, struct `thread` *`td`)

7.69.1 Define Documentation

7.69.1.1 #define `in6pcb` `inpcb`

Definition at line 113 of file `udp6_output.c`.

Referenced by `icmp6_rip6_input()`, `in6_pcbpurgeif0()`, `ip6_raw_ctloutput()`, `sctp6_attach()`, `sctp6_bind()`, `sctp6_connect()`, `sctp6_ctlinput()`, `sctp6_getpeeraddr()`, `sctp6_in6getaddr()`, and `sctp6_send()`.

7.69.1.2 #define `udp6s_opackets` `udps_opackets`

Definition at line 115 of file `udp6_output.c`.

7.69.1.3 #define `udp6stat` `udpstat`

Definition at line 114 of file `udp6_output.c`.

Referenced by `udp6_output()`.

7.69.2 Function Documentation

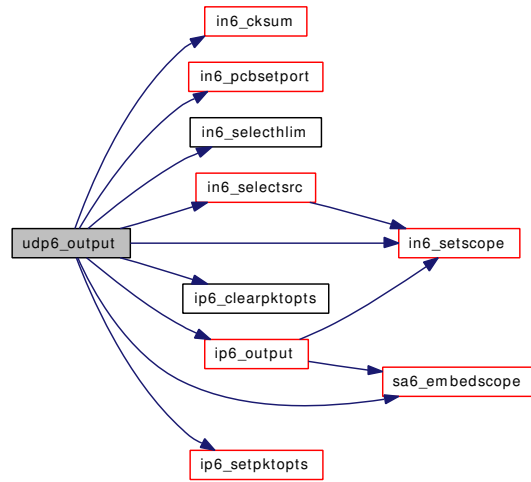
7.69.2.1 int `udp6_output` (struct `in6pcb` *`in6p`, struct `mbuf` *`m`, struct `sockaddr` *`addr6`, struct `mbuf` *`control`, struct `thread` *`td`)

Definition at line 118 of file `udp6_output.c`.

References `in6_cksum()`, `IN6_IS_ADDR_UNSPECIFIED`, `IN6_IS_ADDR_V4MAPPED`, `in6_pcbsetport()`, `in6_selectlim()`, `in6_selectsrc()`, `in6_setscope()`, `ip6_clearpktopts()`, `ip6_output()`, `ip6_setpktopts()`, `ip6_use_defzone`, `sa6_embedscope()`, `sin6`, `sockaddr_in6::sin6_addr`, `sockaddr_in6::sin6_port`, `sockaddr_in6::sin6_scope_id`, and `udp6stat`.

Referenced by `udp6_send()`.

Here is the call graph for this function:

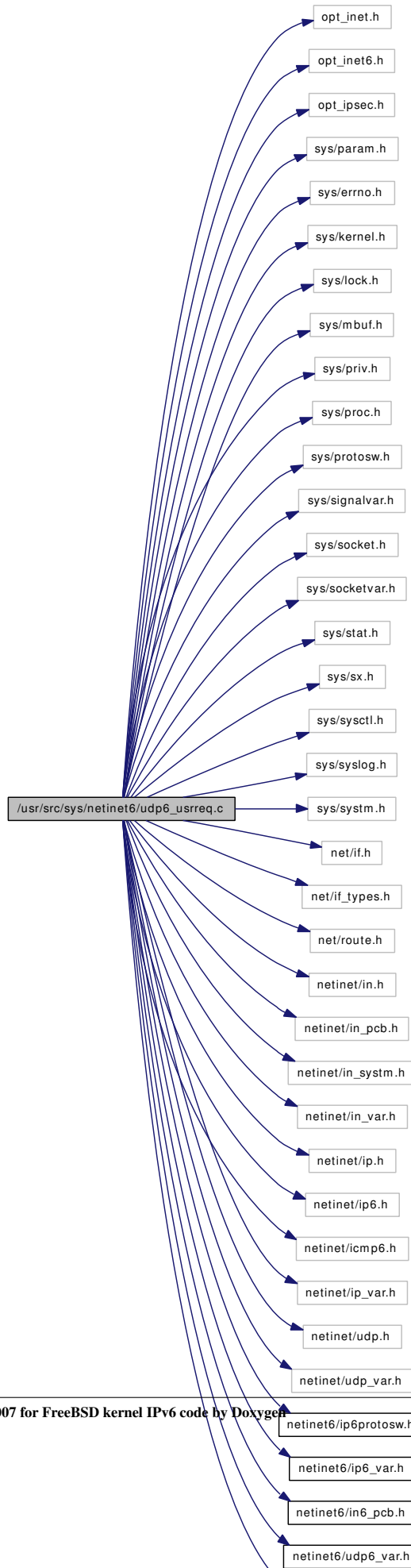


7.70 /usr/src/sys/netinet6/udp6_usrreq.c File Reference

```
#include "opt_inet.h"  
#include "opt_inet6.h"  
#include "opt_ipsec.h"  
#include <sys/param.h>  
#include <sys/errno.h>  
#include <sys/kernel.h>  
#include <sys/lock.h>  
#include <sys/mbuf.h>  
#include <sys/priv.h>  
#include <sys/proc.h>  
#include <sys/protosw.h>  
#include <sys/signalvar.h>  
#include <sys/socket.h>  
#include <sys/socketvar.h>  
#include <sys/stat.h>  
#include <sys/sx.h>  
#include <sys/sysctl.h>  
#include <sys/syslog.h>  
#include <sys/system.h>  
#include <net/if.h>  
#include <net/if_types.h>  
#include <net/route.h>  
#include <netinet/in.h>  
#include <netinet/in_pcb.h>  
#include <netinet/in_system.h>  
#include <netinet/in_var.h>  
#include <netinet/ip.h>  
#include <netinet/ip6.h>  
#include <netinet/icmp6.h>  
#include <netinet/ip_var.h>  
#include <netinet/udp.h>  
#include <netinet/udp_var.h>  
#include <netinet6/ip6protosw.h>  
#include <netinet6/ip6_var.h>  
#include <netinet6/in6_pcb.h>
```

```
#include <netinet6/udp6_var.h>
#include <netinet6/scope6_var.h>
```

Include dependency graph for `udp6_usrreq.c`:



Functions

- static void `udp6_detach` `__P` ((struct socket *so))
- static void `udp6_append` (struct inpcb *in6p, struct mbuf *n, int off, struct [sockaddr_in6](#) *fromsa)
- int `udp6_input` (struct mbuf **mp, int *offp, int proto)
- void `udp6_ctlinput` (int cmd, struct sockaddr *sa, void *d)
- static int `udp6_getcred` (SYSCTL_HANDLER_ARGS)
- `SYSCTL_PROC` (_net_inet6_udp6, OID_AUTO, getcred, CTLTYPE_OPAQUE|CTLFLAG_RW, 0, 0, udp6_getcred, "S,xucred", "Get the xucred of a UDP6 connection")
- static void `udp6_abort` (struct socket *so)
- static int `udp6_attach` (struct socket *so, int proto, struct thread *td)
- static int `udp6_bind` (struct socket *so, struct sockaddr *nam, struct thread *td)
- static void `udp6_close` (struct socket *so)
- static int `udp6_connect` (struct socket *so, struct sockaddr *nam, struct thread *td)
- static void `udp6_detach` (struct socket *so)
- static int `udp6_disconnect` (struct socket *so)
- static int `udp6_send` (struct socket *so, int flags, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td)

Variables

- protosw `inetsw` []
- pr_usrreqs `udp6_usrreqs`

7.70.1 Function Documentation

7.70.1.1 `static void udp6_detach __P` ((struct socket *so)) [static]

7.70.1.2 `SYSCTL_PROC` (_net_inet6_udp6, OID_AUTO, getcred, CTLTYPE_OPAQUE|CTLFLAG_RW, 0, 0, udp6_getcred, "S, xucred", "Get the xucred of a UDP6 connection")

7.70.1.3 `static void udp6_abort` (struct socket *so) [static]

Definition at line 480 of file `udp6_usrreq.c`.

References `IN6_IS_ADDR_UNSPECIFIED`, `in6_pcbdisconnect()`, `in6addr_any`, `inetsw`, and `udbinfo`.

Here is the call graph for this function:



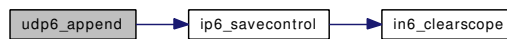
7.70.1.4 `static void udp6_append` (struct inpcb *in6p, struct mbuf *n, int off, struct [sockaddr_in6](#) *fromsa) [static]

Definition at line 124 of file `udp6_usrreq.c`.

References `ipsecstat::in_polvio`, `ip6_savecontrol()`, and `ipsec6stat`.

Referenced by `udp6_input()`.

Here is the call graph for this function:



7.70.1.5 `static int udp6_attach (struct socket * so, int proto, struct thread * td)` [static]

Definition at line 509 of file `udp6_usrreq.c`.

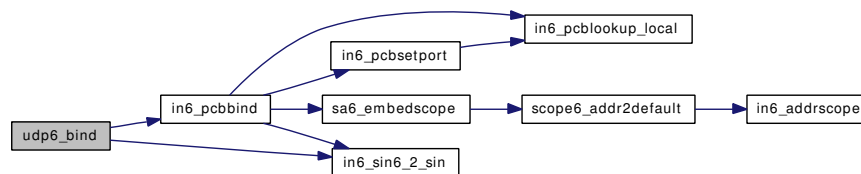
References `udbinfo`.

7.70.1.6 `static int udp6_bind (struct socket * so, struct sockaddr * nam, struct thread * td)` [static]

Definition at line 547 of file `udp6_usrreq.c`.

References `IN6_IS_ADDR_UNSPECIFIED`, `IN6_IS_ADDR_V4MAPPED`, `in6_pcbbind()`, `in6_sin6_2_sin()`, `sockaddr_in6::sin6_addr`, and `udbinfo`.

Here is the call graph for this function:



7.70.1.7 `static void udp6_close (struct socket * so)` [static]

Definition at line 586 of file `udp6_usrreq.c`.

References `IN6_IS_ADDR_UNSPECIFIED`, `in6_pcbdisconnect()`, `in6addr_any`, `inetsw`, and `udbinfo`.

Here is the call graph for this function:

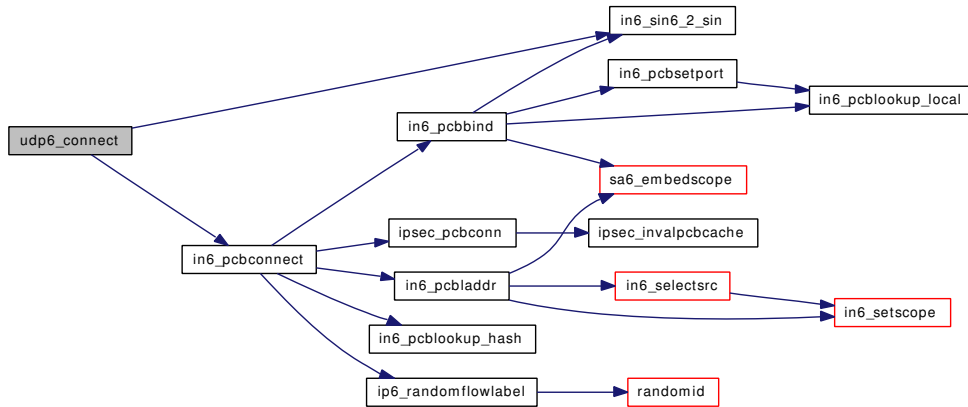


7.70.1.8 `static int udp6_connect (struct socket * so, struct sockaddr * nam, struct thread * td)` [static]

Definition at line 614 of file `udp6_usrreq.c`.

References `IN6_IS_ADDR_UNSPECIFIED`, `IN6_IS_ADDR_V4MAPPED`, `in6_pcbconnect()`, `in6_sin6_2_sin()`, `sockaddr_in6::sin6_addr`, and `udbinfo`.

Here is the call graph for this function:

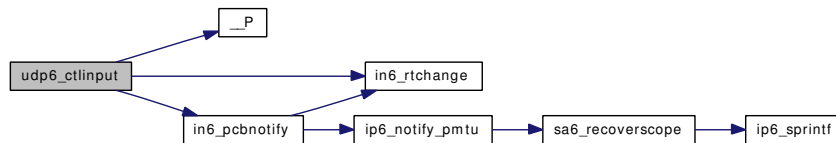


7.70.1.9 void udp6_ctlinput (int cmd, struct sockaddr * sa, void * d)

Definition at line 364 of file udp6_usrreq.c.

References `__P()`, `in6_pcbnotify()`, `in6_rtchange()`, `inet6ctlerrmap`, `ip6ctlparam::ip6c_cmdarg`, `ip6ctlparam::ip6c_ip6`, `ip6ctlparam::ip6c_m`, `ip6ctlparam::ip6c_off`, `ip6ctlparam::ip6c_src`, `sa6_any`, and `udbinfo`.

Here is the call graph for this function:

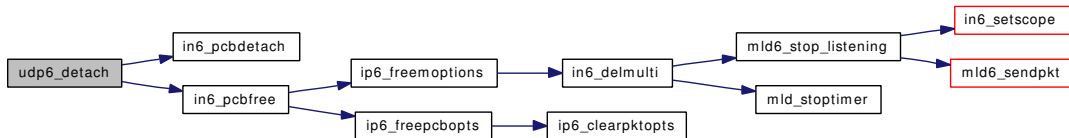


7.70.1.10 static void udp6_detach (struct socket * so) [static]

Definition at line 666 of file udp6_usrreq.c.

References `in6_pcbdetach()`, `in6_pcbfree()`, and `udbinfo`.

Here is the call graph for this function:



7.70.1.11 static int udp6_disconnect (struct socket * so) [static]

Definition at line 681 of file udp6_usrreq.c.

References `IN6_IS_ADDR_UNSPECIFIED`, `in6_pcbdisconnect()`, `in6addr_any`, `inetsw`, and `udbinfo`.

Here is the call graph for this function:

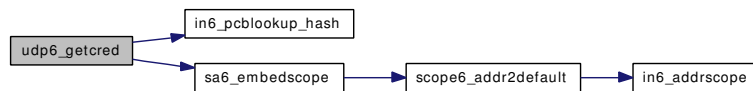


7.70.1.12 static int udp6_getcred (SYSCTL_HANDLER_ARGS) [static]

Definition at line 433 of file udp6_usrreq.c.

References in6_pcblookup_hash(), ip6_use_defzone, sa6_embedscope(), and udbinfo.

Here is the call graph for this function:

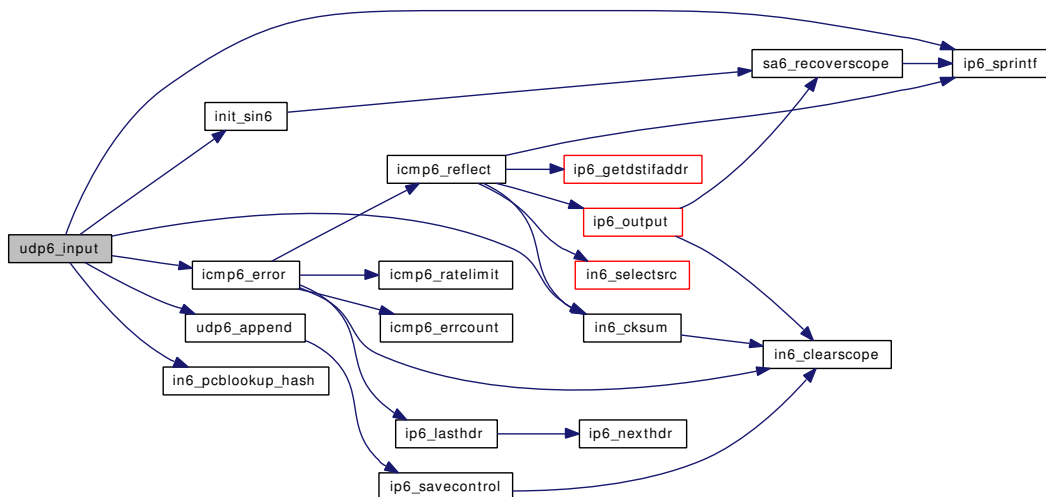


7.70.1.13 int udp6_input (struct mbuf ** mp, int * offp, int proto)

Definition at line 165 of file udp6_usrreq.c.

References faithprefix_p, icmp6_error(), IN6_ARE_ADDR_EQUAL, in6_cksum(), IN6_IS_ADDR_MULTICAST, IN6_IS_ADDR_UNSPECIFIED, in6_pcblookup_hash(), INET6_ADDRSTRLEN, init_sin6(), ip6_sprintf(), sockaddr_in6::sin6_port, udbinfo, and udp6_append().

Here is the call graph for this function:

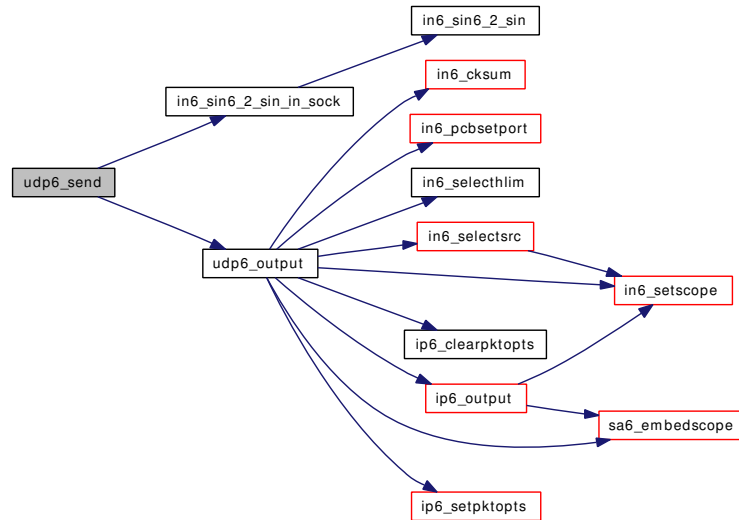


7.70.1.14 static int udp6_send (struct socket * so, int flags, struct mbuf * m, struct sockaddr * addr, struct mbuf * control, struct thread * td) [static]

Definition at line 718 of file udp6_usrreq.c.

References `IN6_IS_ADDR_UNSPECIFIED`, `IN6_IS_ADDR_V4MAPPED`, `in6_sin6_2_sin_in_sock()`, `inetsw`, `sin6`, `sockaddr_in6::sin6_addr`, `udbinfo`, and `udp6_output()`.

Here is the call graph for this function:



7.70.2 Variable Documentation

7.70.2.1 struct protosw `inetsw[]`

7.70.2.2 struct pr_usrreqs `udp6_usrreqs`

Initial value:

```

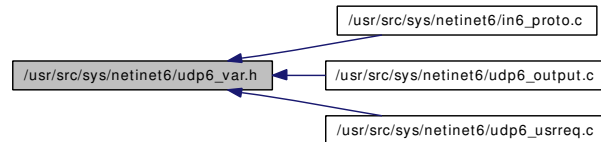
{
    .pru_abort =          udp6_abort,
    .pru_attach =        udp6_attach,
    .pru_bind =          udp6_bind,
    .pru_connect =       udp6_connect,
    .pru_control =       in6_control,
    .pru_detach =        udp6_detach,
    .pru_disconnect =    udp6_disconnect,
    .pru_peeraddr =      in6_mapped_peeraddr,
    .pru_send =          udp6_send,
    .pru_shutdown =      udp_shutdown,
    .pru_sockaddr =      in6_mapped_sockaddr,
    .pru_sosetlabel =    in_pcbsosetLabel,
    .pru_close =         udp6_close
}

```

Definition at line 791 of file `udp6_usrreq.c`.

7.71 /usr/src/sys/netinet6/udp6_var.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- [SYSCTL_DECL](#) ([_net_inet6_udp6](#))
- [void udp6_ctlinput](#) [__P](#) ((int, struct sockaddr *, void *))
- [int udp6_input](#) [__P](#) ((struct mbuf **, int *, int))
- [int udp6_output](#) [__P](#) ((struct inpcb *inp, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td))

Variables

- [pr_usrreqs](#) [udp6_usrreqs](#)

7.71.1 Function Documentation

7.71.1.1 [int udp6_output](#) [__P](#) ((struct inpcb *inp, struct mbuf *m, struct sockaddr *addr, struct mbuf *control, struct thread *td))

7.71.1.2 [int udp6_input](#) [__P](#) ((struct mbuf **, int *, int))

7.71.1.3 [void udp6_ctlinput](#) [__P](#) ((int, struct sockaddr *, void *))

7.71.1.4 [SYSCTL_DECL](#) ([_net_inet6_udp6](#))

7.71.2 Variable Documentation

7.71.2.1 [struct pr_usrreqs](#) [udp6_usrreqs](#)

Definition at line 791 of file [udp6_usrreq.c](#).

Index

/usr/ Directory Reference, 15
/usr/src/ Directory Reference, 13
/usr/src/sys/ Directory Reference, 14
/usr/src/sys/netinet6/ Directory Reference, 11
/usr/src/sys/netinet6/ah.h, 178
/usr/src/sys/netinet6/ah6.h, 180
/usr/src/sys/netinet6/ah_aesxcbcmac.c, 181
/usr/src/sys/netinet6/ah_aesxcbcmac.h, 183
/usr/src/sys/netinet6/ah_core.c, 184
/usr/src/sys/netinet6/ah_input.c, 192
/usr/src/sys/netinet6/ah_output.c, 195
/usr/src/sys/netinet6/dest6.c, 198
/usr/src/sys/netinet6/esp.h, 201
/usr/src/sys/netinet6/esp6.h, 203
/usr/src/sys/netinet6/esp_aesctr.c, 204
/usr/src/sys/netinet6/esp_aesctr.h, 207
/usr/src/sys/netinet6/esp_core.c, 208
/usr/src/sys/netinet6/esp_input.c, 217
/usr/src/sys/netinet6/esp_output.c, 220
/usr/src/sys/netinet6/esp_rijndael.c, 223
/usr/src/sys/netinet6/esp_rijndael.h, 225
/usr/src/sys/netinet6/frag6.c, 226
/usr/src/sys/netinet6/icmp6.c, 233
/usr/src/sys/netinet6/icmp6.h, 245
/usr/src/sys/netinet6/in6.c, 246
/usr/src/sys/netinet6/in6.h, 263
/usr/src/sys/netinet6/in6_cksum.c, 282
/usr/src/sys/netinet6/in6_gif.c, 284
/usr/src/sys/netinet6/in6_gif.h, 289
/usr/src/sys/netinet6/in6_ifattach.c, 290
/usr/src/sys/netinet6/in6_ifattach.h, 299
/usr/src/sys/netinet6/in6_pcb.c, 300
/usr/src/sys/netinet6/in6_pcb.h, 308
/usr/src/sys/netinet6/in6_proto.c, 310
/usr/src/sys/netinet6/in6_rmx.c, 324
/usr/src/sys/netinet6/in6_src.c, 330
/usr/src/sys/netinet6/in6_var.h, 340
/usr/src/sys/netinet6/ip6.h, 358
/usr/src/sys/netinet6/ip6_ecn.h, 359
/usr/src/sys/netinet6/ip6_forward.c, 360
/usr/src/sys/netinet6/ip6_id.c, 363
/usr/src/sys/netinet6/ip6_input.c, 367
/usr/src/sys/netinet6/ip6_mroute.c, 378
/usr/src/sys/netinet6/ip6_mroute.h, 392
/usr/src/sys/netinet6/ip6_output.c, 397
/usr/src/sys/netinet6/ip6_var.h, 411
/usr/src/sys/netinet6/ip6protosw.h, 423
/usr/src/sys/netinet6/ipcomp.h, 424
/usr/src/sys/netinet6/ipcomp6.h, 426
/usr/src/sys/netinet6/ipcomp_core.c, 427
/usr/src/sys/netinet6/ipcomp_input.c, 432
/usr/src/sys/netinet6/ipcomp_output.c, 435
/usr/src/sys/netinet6/ipsec.c, 438
/usr/src/sys/netinet6/ipsec.h, 456
/usr/src/sys/netinet6/ipsec6.h, 465
/usr/src/sys/netinet6/mld6.c, 468
/usr/src/sys/netinet6/mld6_var.h, 475
/usr/src/sys/netinet6/nd6.c, 478
/usr/src/sys/netinet6/nd6.h, 495
/usr/src/sys/netinet6/nd6_nbr.c, 509
/usr/src/sys/netinet6/nd6_rtr.c, 519
/usr/src/sys/netinet6/pim6.h, 535
/usr/src/sys/netinet6/pim6_var.h, 536
/usr/src/sys/netinet6/raw_ip6.c, 537
/usr/src/sys/netinet6/raw_ip6.h, 545
/usr/src/sys/netinet6/route6.c, 546
/usr/src/sys/netinet6/scope6.c, 549
/usr/src/sys/netinet6/scope6_var.h, 555
/usr/src/sys/netinet6/sctp6_usrreq.c, 557
/usr/src/sys/netinet6/sctp6_var.h, 563
/usr/src/sys/netinet6/tcp6_var.h, 564
/usr/src/sys/netinet6/udp6_output.c, 565
/usr/src/sys/netinet6/udp6_usrreq.c, 569
/usr/src/sys/netinet6/udp6_var.h, 577
__FBSDID
 sctp6_usrreq.c, 559
 sctp6_var.h, 563
__KAME_VERSION
 in6.h, 266
__KAME__
 in6.h, 266
__P
 ah.h, 179
 ah6.h, 180
 ah_aesxcbcmac.h, 183
 ah_algorithm, 22
 ah_core.c, 187
 esp.h, 202
 esp6.h, 203
 esp_aesctr.h, 207

- esp_algorithm, 29
- esp_core.c, 212
- esp_output.c, 222
- esp_rijndael.h, 225
- frag6.c, 229
- icmp6.c, 236
- in6.c, 249
- in6.h, 280
- in6_gif.h, 289
- in6_ifattach.c, 293
- in6_ifattach.h, 299
- in6_pcb.h, 309
- in6_proto.c, 318
- in6_rmx.c, 327
- in6_src.c, 335
- in6_var.h, 355
- ip6_input.c, 370
- ip6_mroute.c, 384
- ip6_mroute.h, 396
- ip6_output.c, 402
- ip6_var.h, 418
- ip6protosw, 102
- ipcomp.h, 425
- ipcomp6.h, 426
- ipcomp_algorithm, 113
- ipcomp_core.c, 429
- ipcomp_output.c, 437
- ipsec.c, 443
- ipsec.h, 464
- ipsec6.h, 466
- nd6.c, 482
- nd6.h, 506
- nd6_nbr.c, 512
- nd6_rtr.c, 523
- pim6_var.h, 536
- route6.c, 547
- scope6_var.h, 556
- sctp6_var.h, 563
- tcp6_var.h, 564
- udp6_usrreq.c, 572
- udp6_var.h, 577
- __u6_addr
 - in6_addr, 37
- __u6_addr16
 - in6_addr, 37
- __u6_addr32
 - in6_addr, 37
- __u6_addr8
 - in6_addr, 37
- add_addrsel_policyent
 - in6_src.c, 335
- add_m6fc
 - ip6_mroute.c, 384
- add_m6if
 - ip6_mroute.c, 384
- ADDCARRY
 - in6_cksum.c, 282
- addr
 - in6_addrpolicy, 39
 - in6_nbrinfo, 61
- ADDR_LABEL_NOTAPP
 - in6_src.c, 333
- addrmask
 - in6_addrpolicy, 39
- ADDRSEL_LOCK
 - in6_src.c, 333
- addrsel_lock
 - in6_src.c, 339
- ADDRSEL_LOCK_ASSERT
 - in6_src.c, 333
- ADDRSEL_LOCK_INIT
 - in6_src.c, 333
- addrsel_policy_init
 - in6_src.c, 335
- addrsel_policyent, 17
- addrsel_policytab
 - in6_src.c, 339
- ADDRSEL_SLOCK
 - in6_src.c, 333
- ADDRSEL_SUNLOCK
 - in6_src.c, 333
- addrsel_sxlock
 - in6_src.c, 339
- ADDRSEL_SXLOCK_INIT
 - in6_src.c, 333
- ADDRSEL_UNLOCK
 - in6_src.c, 334
- ADDRSEL_XLOCK
 - in6_src.c, 334
- ADDRSEL_XUNLOCK
 - in6_src.c, 334
- advrtr
 - in6_oprlist, 67
 - in6_prlist, 78
- advrtrs
 - in6_oprlist, 67
 - in6_prefix, 70
 - in6_prlist, 78
- AES_BLOCKSIZE
 - ah_aesxcbcmac.c, 182
 - esp_aesctr.c, 205
- aesctr_ctx, 18
 - r_ek, 18
 - r_nr, 18
- aesxcbc_ctx, 19
 - buf, 19
 - buflen, 19

- e, 19
 - k2, 19
 - k3, 19
 - r_k1s, 19
 - r_k2s, 20
 - r_k3s, 20
 - r_nr, 20
 - ah, 21
 - ah_len, 21
 - ah_nxt, 21
 - ah_reserve, 21
 - ah_spi, 21
 - ah.h
 - __P, 179
 - AH_MAXSUMSIZE, 179
 - ah6.h
 - __P, 180
 - ah_aes_xcbc_mac_init
 - ah_aesxcbcmac.c, 182
 - ah_aes_xcbc_mac_loop
 - ah_aesxcbcmac.c, 182
 - ah_aes_xcbc_mac_result
 - ah_aesxcbcmac.c, 182
 - ah_aesxcbcmac.c
 - AES_BLOCKSIZE, 182
 - ah_aes_xcbc_mac_init, 182
 - ah_aes_xcbc_mac_loop, 182
 - ah_aes_xcbc_mac_result, 182
 - ah_aesxcbcmac.h
 - __P, 183
 - ah_algorithm, 22
 - __P, 22
 - keymax, 22
 - keymin, 22
 - name, 22
 - ah_algorithm_lookup
 - ah_core.c, 187
 - ah_algorithm_state, 24
 - foo, 24
 - sav, 24
 - ah_algorithms
 - ah_core.c, 191
 - ah_common_mature
 - ah_core.c, 187
 - ah_core.c
 - __P, 187
 - ah_algorithm_lookup, 187
 - ah_algorithms, 191
 - ah_common_mature, 187
 - ah_hmac_md5_init, 188
 - ah_hmac_md5_loop, 188
 - ah_hmac_md5_result, 188
 - ah_hmac_ripemd160_init, 188
 - ah_hmac_ripemd160_loop, 188
 - ah_hmac_ripemd160_result, 188
 - ah_hmac_sha1_init, 188
 - ah_hmac_sha1_loop, 188
 - ah_hmac_sha1_result, 189
 - ah_hmac_sha2_256_init, 189
 - ah_hmac_sha2_256_loop, 189
 - ah_hmac_sha2_256_result, 189
 - ah_hmac_sha2_384_init, 189
 - ah_hmac_sha2_384_loop, 189
 - ah_hmac_sha2_384_result, 189
 - ah_hmac_sha2_512_init, 189
 - ah_hmac_sha2_512_loop, 189
 - ah_hmac_sha2_512_result, 190
 - ah_keyed_md5_init, 190
 - ah_keyed_md5_loop, 190
 - ah_keyed_md5_mature, 190
 - ah_keyed_md5_result, 190
 - ah_keyed_sha1_init, 190
 - ah_keyed_sha1_loop, 190
 - ah_keyed_sha1_result, 190
 - ah_none_init, 190
 - ah_none_loop, 191
 - ah_none_mature, 191
 - ah_none_result, 191
 - ah_sumsiz_1216, 191
 - ah_sumsiz_zero, 191
 - ah_update_mbuf, 191
 - MD5_RESULTLEN, 187
 - RIPEND160_RESULTLEN, 187
- ah_hdrlen
 - ah_output.c, 196
 - ah_hdrsiz
 - ah_output.c, 197
 - ah_hmac_md5_init
 - ah_core.c, 188
 - ah_hmac_md5_loop
 - ah_core.c, 188
 - ah_hmac_md5_result
 - ah_core.c, 188
 - ah_hmac_ripemd160_init
 - ah_core.c, 188
 - ah_hmac_ripemd160_loop
 - ah_core.c, 188
 - ah_hmac_ripemd160_result
 - ah_core.c, 188
 - ah_hmac_sha1_init
 - ah_core.c, 188
 - ah_hmac_sha1_loop
 - ah_core.c, 188
 - ah_hmac_sha1_result
 - ah_core.c, 189
 - ah_hmac_sha2_256_init
 - ah_core.c, 189
 - ah_hmac_sha2_256_loop

- ah_core.c, 189
- ah_hmac_sha2_256_result
 - ah_core.c, 189
- ah_hmac_sha2_384_init
 - ah_core.c, 189
- ah_hmac_sha2_384_loop
 - ah_core.c, 189
- ah_hmac_sha2_384_result
 - ah_core.c, 189
- ah_hmac_sha2_512_init
 - ah_core.c, 189
- ah_hmac_sha2_512_loop
 - ah_core.c, 189
- ah_hmac_sha2_512_result
 - ah_core.c, 190
- ah_input.c
 - IPLen_FLIPPED, 194
 - KEYDEBUG, 194
- ah_keyed_md5_init
 - ah_core.c, 190
- ah_keyed_md5_loop
 - ah_core.c, 190
- ah_keyed_md5_mature
 - ah_core.c, 190
- ah_keyed_md5_result
 - ah_core.c, 190
- ah_keyed_sha1_init
 - ah_core.c, 190
- ah_keyed_sha1_loop
 - ah_core.c, 190
- ah_keyed_sha1_result
 - ah_core.c, 190
- ah_len
 - ah, 21
 - newah, 150
- AH_MAXSUMSIZE
 - ah.h, 179
- ah_none_init
 - ah_core.c, 190
- ah_none_loop
 - ah_core.c, 191
- ah_none_mature
 - ah_core.c, 191
- ah_none_result
 - ah_core.c, 191
- ah_nxt
 - ah, 21
 - newah, 150
- ah_output.c
 - ah_hdrlen, 196
 - ah_hdrsiz, 197
- ah_reserve
 - ah, 21
 - newah, 150
- ah_seq
 - newah, 150
- ah_spi
 - ah, 21
 - newah, 150
- ah_sumsiz_1216
 - ah_core.c, 191
- ah_sumsiz_zero
 - ah_core.c, 191
- ah_update_mbuf
 - ah_core.c, 191
- all1_sa
 - nd6.c, 492
- asked
 - in6_nbrinfo, 61
- autonomous
 - in6_prflags::prf_ra, 76
 - in6_rrenumreq::irr_raflagmask, 82
- basereachable
 - in6_ondireq, 65
 - nd_ifinfo, 140
- BREAK
 - in6_src.c, 334
- buf
 - aesxcbc_ctx, 19
- buflen
 - aesxcbc_ctx, 19
- bytecnt
 - sioc_sg_req6, 172
- cache
 - inpcbpolicy, 85
- cacheflags
 - inpcbpolicy, 85
- cachegen
 - inpcbpolicy, 85
- cacheidx
 - inpcbpolicy, 85
- cast128_decrypt
 - esp_core.c, 211
- cast128_encrypt
 - esp_core.c, 211
- cast128_key
 - esp_core.c, 211
- cast128_setkey
 - esp_core.c, 211
- cblock, 25
 - cblock, 25
 - ctr, 25
 - iv, 25
 - nonce, 25
 - v, 25
- chlim

- in6_ondireq, 65
- nd_ifinfo, 140
- clear_llinfo_pqueue
 - nd6.c, 482
- comp_cpi
 - ipcomp, 112
- comp_flags
 - ipcomp, 112
- comp_nxt
 - ipcomp, 112
- copypktopts
 - ip6_output.c, 402
- count
 - secspacq, 170
- created
 - secpolicy, 167
 - secspacq, 170
- ctr
 - cblock, 25
- dad_ignore_ns
 - nd6_nbr.c, 518
- dad_init
 - nd6_nbr.c, 518
- dad_maxtry
 - nd6_nbr.c, 518
- dadq, 26
 - nd6_nbr.c, 518
- decrprefd
 - in6_prflags::prf_rr, 77
- decrvalid
 - in6_prflags::prf_rr, 77
- DEF_TEMP_PREFERRED_LIFETIME
 - nd6.h, 498
- DEF_TEMP_VALID_LIFETIME
 - nd6.h, 498
- defaultaddrpolicy
 - in6_src.c, 339
- deflate_alloc
 - ipcomp_core.c, 429
- deflate_common
 - ipcomp_core.c, 429
- deflate_compress
 - ipcomp_core.c, 429
- deflate_decompress
 - ipcomp_core.c, 430
- deflate_free
 - ipcomp_core.c, 430
- deflate_memlevel
 - ipcomp_core.c, 430
- deflate_policy
 - ipcomp_core.c, 430
- deflate_window_in
 - ipcomp_core.c, 430
- deflate_window_out
 - ipcomp_core.c, 430
- defrouter
 - in6_drlist, 45
- defrouter_addreq
 - nd6_rtr.c, 523
- defrouter_delreq
 - nd6_rtr.c, 523
- defrouter_lookup
 - nd6_rtr.c, 523
- defrouter_reset
 - nd6_rtr.c, 524
- defrouter_select
 - nd6_rtr.c, 524
- defrtrlist_del
 - nd6_rtr.c, 524
- defrtrlist_update
 - nd6_rtr.c, 525
- del_m6fc
 - ip6_mroute.c, 384
- del_m6if
 - ip6_mroute.c, 385
- delete_addrsel_policyent
 - in6_src.c, 335
- dest6.c
 - dest6_input, 199
- dest6_input
 - dest6.c, 199
- digits
 - in6.c, 260
- dir
 - secpolicy, 167
- DOMAIN_SET
 - in6_proto.c, 318
- done
 - nd_opts, 142
- draining
 - rtqk_arg, 163
- DRLSTSIZ
 - nd6.h, 498
- dst
 - ipsec_output_state, 115
 - secpolicyindex, 169
- dump_addrsel_policyent
 - in6_src.c, 336
- e
 - aesxcbc_ctx, 19
- elen
 - ip6_output.c, 400
- encap
 - ipsec_output_state, 115
- ENCAP_HOPS
 - ip6_mroute.c, 381

- esp, 27
 - esp_spi, 27
- esp.h
 - __P, 202
- esp6.h
 - __P, 203
- esp_3des_blockdecrypt
 - esp_core.c, 212
- esp_3des_blockencrypt
 - esp_core.c, 212
- esp_3des_schedlen
 - esp_core.c, 212
- esp_3des_schedule
 - esp_core.c, 212
- esp_aesctr.c
 - AES_BLOCKSIZE, 205
 - esp_aesctr_decrypt, 205
 - esp_aesctr_encrypt, 205
 - esp_aesctr_mature, 205
 - esp_aesctr_schedlen, 206
 - esp_aesctr_schedule, 206
 - NONCESIZE, 205
- esp_aesctr.h
 - __P, 207
- esp_aesctr_decrypt
 - esp_aesctr.c, 205
- esp_aesctr_encrypt
 - esp_aesctr.c, 205
- esp_aesctr_mature
 - esp_aesctr.c, 205
- esp_aesctr_schedlen
 - esp_aesctr.c, 206
- esp_aesctr_schedule
 - esp_aesctr.c, 206
- esp_algorithm, 28
 - __P, 29
 - ivlenval, 29
 - keymax, 29
 - keymin, 29
 - name, 29
 - padbound, 29
- esp_algorithm_lookup
 - esp_core.c, 212
- esp_algorithms
 - esp_core.c, 216
- esp_auth
 - esp_core.c, 212
- esp_blowfish_blockdecrypt
 - esp_core.c, 213
- esp_blowfish_blockencrypt
 - esp_core.c, 213
- esp_blowfish_schedlen
 - esp_core.c, 213
- esp_blowfish_schedule
 - esp_core.c, 213
- esp_core.c, 213
 - esp_cast128_blockdecrypt
 - esp_core.c, 213
 - esp_cast128_blockencrypt
 - esp_core.c, 213
 - esp_cast128_schedlen
 - esp_core.c, 213
 - esp_cast128_schedule
 - esp_core.c, 213
 - esp_cbc_decrypt
 - esp_core.c, 214
 - esp_cbc_encrypt
 - esp_core.c, 214
 - esp_cbc_mature
 - esp_core.c, 214
 - esp_common_ivlen
 - esp_core.c, 214
 - esp_core.c
 - __P, 212
 - cast128_decrypt, 211
 - cast128_encrypt, 211
 - cast128_key, 211
 - cast128_setkey, 211
 - esp_3des_blockdecrypt, 212
 - esp_3des_blockencrypt, 212
 - esp_3des_schedlen, 212
 - esp_3des_schedule, 212
 - esp_algorithm_lookup, 212
 - esp_algorithms, 216
 - esp_auth, 212
 - esp_blowfish_blockdecrypt, 213
 - esp_blowfish_blockencrypt, 213
 - esp_blowfish_schedlen, 213
 - esp_blowfish_schedule, 213
 - esp_cast128_blockdecrypt, 213
 - esp_cast128_blockencrypt, 213
 - esp_cast128_schedlen, 213
 - esp_cast128_schedule, 213
 - esp_cbc_decrypt, 214
 - esp_cbc_encrypt, 214
 - esp_cbc_mature, 214
 - esp_common_ivlen, 214
 - esp_des_blockdecrypt, 214
 - esp_des_blockencrypt, 214
 - esp_des_schedlen, 214
 - esp_des_schedule, 215
 - esp_descbc_ivlen, 215
 - esp_descbc_mature, 215
 - esp_max_ivlen, 215
 - esp_null_decrypt, 215
 - esp_null_encrypt, 215
 - esp_null_mature, 215
 - esp_schedule, 215
 - MAXIVLEN, 211

- esp_des_blockdecrypt
 - esp_core.c, 214
- esp_des_blockencrypt
 - esp_core.c, 214
- esp_des_schedlen
 - esp_core.c, 214
- esp_des_schedule
 - esp_core.c, 215
- esp_descbc_ivlen
 - esp_core.c, 215
- esp_descbc_mature
 - esp_core.c, 215
- esp_hdrsiz
 - esp_output.c, 222
- esp_input.c
 - ESPMAXLEN, 219
 - IPLen_FLIPPED, 219
- esp_max_ivlen
 - esp_core.c, 215
- esp_null_decrypt
 - esp_core.c, 215
- esp_null_encrypt
 - esp_core.c, 215
- esp_null_mature
 - esp_core.c, 215
- esp_nxt
 - esptail, 31
- esp_output
 - esp_output.c, 222
- esp_output.c
 - __P, 222
 - esp_hdrsiz, 222
 - esp_output, 222
- esp_padlen
 - esptail, 31
- esp_rijndael.c
 - esp_rijndael_blockdecrypt, 224
 - esp_rijndael_blockencrypt, 224
 - esp_rijndael_schedlen, 224
 - esp_rijndael_schedule, 224
- esp_rijndael.h
 - __P, 225
- esp_rijndael_blockdecrypt
 - esp_rijndael.c, 224
- esp_rijndael_blockencrypt
 - esp_rijndael.c, 224
- esp_rijndael_schedlen
 - esp_rijndael.c, 224
- esp_rijndael_schedule
 - esp_rijndael.c, 224
- esp_schedule
 - esp_core.c, 215
- esp_seq
 - newesp, 151
- esp_spi
 - esp, 27
 - newesp, 151
- ESPMAXLEN
 - esp_input.c, 219
- esptail, 31
 - esp_nxt, 31
 - esp_padlen, 31
- EUI64_GBIT
 - in6_ifattach.c, 292
- EUI64_GROUP
 - in6_ifattach.c, 292
- EUI64_INDIVIDUAL
 - in6_ifattach.c, 292
- EUI64_LOCAL
 - in6_ifattach.c, 292
- EUI64_TO_IFID
 - in6_ifattach.c, 292
- EUI64_UBIT
 - in6_ifattach.c, 293
- EUI64_UNIVERSAL
 - in6_ifattach.c, 293
- expire
 - in6_defrouter, 43
 - in6_drlist, 45
 - in6_nbrinfo, 61
 - in6_oprlist, 67
 - in6_prefix, 70
 - in6_prlist, 78
 - nd_defrouter, 138
- EXPIRE_TIMEOUT
 - ip6_mroute.c, 381
- expire_upcalls
 - ip6_mroute.c, 385
- expire_upcalls_ch
 - ip6_mroute.c, 390
- faithprefix_p
 - in6.c, 260
 - in6.h, 280
- find_pfxlist_reachable_router
 - nd6_rtr.c, 525
- flags
 - in6_defrouter, 43
 - in6_drlist, 45
 - in6_ondireq, 65
 - in6_prefix, 70
 - nd_defrouter, 138
 - nd_ifinfo, 140
- foo
 - ah_algorithm_state, 24
- found
 - rtqk_arg, 163
- frag6.c

- __P, 229
- frag6_change, 229
- frag6_deq, 229
- frag6_drain, 229
- frag6_enq, 229
- frag6_freef, 230
- frag6_init, 230
- frag6_input, 230
- frag6_insque, 231
- frag6_nfragpackets, 232
- frag6_nfrags, 232
- frag6_remque, 231
- frag6_slowtimo, 231
- IN6_IFSTAT_STRICT, 228
- ip6q, 232
- IP6Q_LOCK, 228
- IP6Q_LOCK_ASSERT, 228
- IP6Q_LOCK_INIT, 228
- IP6Q_TRYLOCK, 228
- IP6Q_UNLOCK, 228
- ip6qlock, 232
- MALLOC_DEFINE, 232
- frag6_change
 - frag6.c, 229
- frag6_deq
 - frag6.c, 229
- frag6_drain
 - frag6.c, 229
- frag6_enq
 - frag6.c, 229
- frag6_freef
 - frag6.c, 230
- frag6_init
 - frag6.c, 230
- frag6_input
 - frag6.c, 230
- frag6_insque
 - frag6.c, 231
- frag6_nfragpackets
 - frag6.c, 232
- frag6_nfrags
 - frag6.c, 232
- frag6_remque
 - frag6.c, 231
- frag6_slowtimo
 - frag6.c, 231
- generate_tmp_ifid
 - in6_ifattach.c, 293
- get_ifid
 - in6_ifattach.c, 293
- get_mif6_cnt
 - ip6_mroute.c, 385
- get_rand_ifid
 - in6_ifattach.c, 294
- get_sg_cnt
 - ip6_mroute.c, 385
- GET_TIME
 - ip6_mroute.h, 393
- gif_encapcheck6
 - in6_gif.c, 286
- GIF_HLIM
 - in6_gif.h, 289
- gif_validate6
 - in6_gif.c, 286
- grp
 - sioc_sg_req6, 172
- hdrs
 - ipsecaux, 116
- hostnamelen
 - icmp6.c, 236
 - in6.c, 249
- howmany
 - ip6_mroute.h, 393
- i6mm_maddr
 - in6_multi_mship, 59
- i_ia
 - in6_multistep, 60
- i_in6m
 - in6_multistep, 60
- ia62ifa
 - in6.c, 249
- ia6_createtime
 - in6_ifaddr, 47
- IA6_DSTIN6
 - in6_var.h, 343
- IA6_DSTSIN6
 - in6_var.h, 343
- ia6_flags
 - in6_ifaddr, 47
- IA6_IN6
 - in6_var.h, 343
- ia6_lifetime
 - in6_ifaddr, 48
- IA6_MASKIN6
 - in6_var.h, 343
- ia6_ndpr
 - in6_ifaddr, 48
- IA6_SIN6
 - in6_var.h, 343
- ia6_updatetime
 - in6_ifaddr, 48
- ia6t_expire
 - in6_addrlifetime, 38
- ia6t_pltime
 - in6_addrlifetime, 38

- ia6t_preferred
 - in6_addrlifetime, 38
- ia6t_vltime
 - in6_addrlifetime, 38
- ia_addr
 - in6_ifaddr, 48
- ia_dstaddr
 - in6_ifaddr, 48
- ia_flags
 - in6_var.h, 343
- ia_ifa
 - in6_ifaddr, 48
- ia_ifp
 - in6_var.h, 344
- ia_net
 - in6_ifaddr, 48
- ia_next
 - in6_ifaddr, 48
- ia_plen
 - in6_ifaddr, 49
- ia_prefixmask
 - in6_ifaddr, 49
- ibytes
 - sioc_mif_req6, 171
- icmp6.c
 - __P, 236
 - hostnamelen, 236
 - icmp6_ctloutput, 236
 - icmp6_errcount, 236
 - icmp6_error, 236
 - icmp6_error2, 237
 - icmp6_fasttimo, 237
 - icmp6_init, 238
 - icmp6_input, 238
 - icmp6_mtudisc_update, 238
 - icmp6_nodeinfo, 243
 - icmp6_notify_error, 239
 - icmp6_ratelimit, 239
 - icmp6_redirect_diag, 239
 - icmp6_redirect_input, 240
 - icmp6_redirect_output, 240
 - icmp6_reflect, 241
 - icmp6_rip6_input, 241
 - icmp6errpps_count, 243
 - icmp6errppslim, 243
 - icmp6errppslim_last, 243
 - icmp6stat, 243
 - inet6domain, 243
 - ni6_addrs, 241
 - ni6_dnsmatch, 242
 - ni6_input, 242
 - ni6_nametodns, 242
 - ni6_store_addrs, 242
 - ripcb, 244
 - ripcbinfo, 244
- icmp6_ctloutput
 - icmp6.c, 236
- icmp6_errcount
 - icmp6.c, 236
- icmp6_error
 - icmp6.c, 236
- icmp6_error2
 - icmp6.c, 237
- icmp6_fasttimo
 - icmp6.c, 237
- ICMP6_FILTER
 - in6.h, 266
- icmp6_ifstat, 32
- ifs6_in_adminprohib, 32
- ifs6_in_dstunreach, 32
- ifs6_in_echo, 33
- ifs6_in_echoreply, 33
- ifs6_in_error, 33
- ifs6_in_mlddone, 33
- ifs6_in_mldquery, 33
- ifs6_in_mldreport, 33
- ifs6_in_msg, 33
- ifs6_in_neighboradvert, 33
- ifs6_in_neighborsolicit, 33
- ifs6_in_paramprob, 33
- ifs6_in_pkttoobig, 33
- ifs6_in_redirect, 34
- ifs6_in_routeradvert, 34
- ifs6_in_routersolicit, 34
- ifs6_in_timeexceed, 34
- ifs6_out_adminprohib, 34
- ifs6_out_dstunreach, 34
- ifs6_out_echo, 34
- ifs6_out_echoreply, 34
- ifs6_out_error, 34
- ifs6_out_mlddone, 34
- ifs6_out_mldquery, 34
- ifs6_out_mldreport, 35
- ifs6_out_msg, 35
- ifs6_out_neighboradvert, 35
- ifs6_out_neighborsolicit, 35
- ifs6_out_paramprob, 35
- ifs6_out_pkttoobig, 35
- ifs6_out_redirect, 35
- ifs6_out_routeradvert, 35
- ifs6_out_routersolicit, 35
- ifs6_out_timeexceed, 35
- in6_ifextra, 50
- icmp6_init
 - icmp6.c, 238
- icmp6_input
 - icmp6.c, 238
- icmp6_mtudisc_update

- icmp6.c, 238
- icmp6_nodeinfo
 - icmp6.c, 243
 - in6_proto.c, 319
- icmp6_notify_error
 - icmp6.c, 239
- icmp6_ratelimit
 - icmp6.c, 239
- icmp6_rediraccept
 - in6_proto.c, 319
- icmp6_redirect_diag
 - icmp6.c, 239
- icmp6_redirect_input
 - icmp6.c, 240
- icmp6_redirect_output
 - icmp6.c, 240
- icmp6_redirtimeout
 - in6_proto.c, 320
- icmp6_reflect
 - icmp6.c, 241
- icmp6_rip6_input
 - icmp6.c, 241
- icmp6errpps_count
 - icmp6.c, 243
- icmp6errppslim
 - icmp6.c, 243
 - in6_proto.c, 320
- icmp6errppslim_last
 - icmp6.c, 243
- icmp6stat
 - icmp6.c, 243
 - in6_var.h, 356
- icount
 - sioc_mif_req6, 171
- id
 - secpolicy, 167
- IF_CLR
 - ip6_mroute.h, 393
- IF_COPY
 - ip6_mroute.h, 393
- if_index
 - in6_defrouter, 43
 - in6_drlist, 45
 - in6_oprlist, 67
 - in6_prefix, 70
 - in6_prlist, 78
- IF_ISSET
 - ip6_mroute.h, 393
- if_mask
 - ip6_mroute.h, 396
- IF_SET
 - ip6_mroute.h, 393
- if_set, 36
 - ifs_bits, 36
- IF_SETSIZE
 - ip6_mroute.h, 393
- IF_ZERO
 - ip6_mroute.h, 393
- ifa2ia6
 - in6.c, 249
- IFA6_IS_DEPRECATED
 - in6.h, 266
- IFA6_IS_INVALID
 - in6.h, 266
- IFA_DSTIN6
 - in6_var.h, 344
- IFA_IN6
 - in6_var.h, 344
- ifatoia6
 - in6.h, 266
 - in6_pcb.h, 308
 - raw_ip6.c, 539
- IFID_LOCAL
 - in6_ifattach.c, 293
- IFID_UNIVERSAL
 - in6_ifattach.c, 293
- ifindex
 - in6_ndifreq, 63
- ifname
 - in6_drlist, 45
 - in6_nbrinfo, 61
 - in6_ndifreq, 63
 - in6_ndireq, 64
 - in6_ondireq, 65
 - in6_oprlist, 68
 - in6_prlist, 79
- ifp
 - nd_defrouter, 138
 - rtdetq, 162
- IFP_TO_IA6
 - in6_var.h, 344
- IFPR_IN6
 - in6_var.h, 344
- ifr_ifru
 - in6_ifreq, 52
- ifr_name
 - in6_ifreq, 52
- ifra_addr
 - in6_aliasreq, 41
- ifra_dstaddr
 - in6_aliasreq, 41
- ifra_flags
 - in6_aliasreq, 41
- ifra_lifetime
 - in6_aliasreq, 41
- ifra_name
 - in6_aliasreq, 42
- ifra_prefixmask

- in6_aliasreq, 42
- ifru_addr
 - in6_ifreq, 52
- ifru_data
 - in6_ifreq, 52
- ifru_dstaddr
 - in6_ifreq, 53
- ifru_flags
 - in6_ifreq, 53
- ifru_flags6
 - in6_ifreq, 53
- ifru_icmp6stat
 - in6_ifreq, 53
- ifru_lifetime
 - in6_ifreq, 53
- ifru_metric
 - in6_ifreq, 53
- ifru_scope_id
 - in6_ifreq, 53
- ifru_stat
 - in6_ifreq, 53
- ifs6_in_addrerr
 - in6_ifstat, 54
- ifs6_in_adminprohib
 - icmp6_ifstat, 32
- ifs6_in_deliver
 - in6_ifstat, 54
- ifs6_in_discard
 - in6_ifstat, 54
- ifs6_in_dstunreach
 - icmp6_ifstat, 32
- ifs6_in_echo
 - icmp6_ifstat, 33
- ifs6_in_echoreply
 - icmp6_ifstat, 33
- ifs6_in_error
 - icmp6_ifstat, 33
- ifs6_in_hdrerr
 - in6_ifstat, 54
- ifs6_in_mcast
 - in6_ifstat, 54
- ifs6_in_mlddone
 - icmp6_ifstat, 33
- ifs6_in_mldquery
 - icmp6_ifstat, 33
- ifs6_in_mldreport
 - icmp6_ifstat, 33
- ifs6_in_msg
 - icmp6_ifstat, 33
- ifs6_in_neighboradvert
 - icmp6_ifstat, 33
- ifs6_in_neighborsolicit
 - icmp6_ifstat, 33
- ifs6_in_noroute
 - in6_ifstat, 55
- ifs6_in_paramprob
 - icmp6_ifstat, 33
- ifs6_in_pkttoobig
 - icmp6_ifstat, 33
- ifs6_in_protounknown
 - in6_ifstat, 55
- ifs6_in_receive
 - in6_ifstat, 55
- ifs6_in_redirect
 - icmp6_ifstat, 34
- ifs6_in_routeradvert
 - icmp6_ifstat, 34
- ifs6_in_routersolicit
 - icmp6_ifstat, 34
- ifs6_in_timeexceed
 - icmp6_ifstat, 34
- ifs6_in_toobig
 - in6_ifstat, 55
- ifs6_in_truncated
 - in6_ifstat, 55
- ifs6_out_adminprohib
 - icmp6_ifstat, 34
- ifs6_out_discard
 - in6_ifstat, 55
- ifs6_out_dstunreach
 - icmp6_ifstat, 34
- ifs6_out_echo
 - icmp6_ifstat, 34
- ifs6_out_echoreply
 - icmp6_ifstat, 34
- ifs6_out_error
 - icmp6_ifstat, 34
- ifs6_out_forward
 - in6_ifstat, 55
- ifs6_out_fragcreat
 - in6_ifstat, 55
- ifs6_out_fragfail
 - in6_ifstat, 55
- ifs6_out_fragok
 - in6_ifstat, 55
- ifs6_out_mcast
 - in6_ifstat, 55
- ifs6_out_mlddone
 - icmp6_ifstat, 34
- ifs6_out_mldquery
 - icmp6_ifstat, 34
- ifs6_out_mldreport
 - icmp6_ifstat, 35
- ifs6_out_msg
 - icmp6_ifstat, 35
- ifs6_out_neighboradvert
 - icmp6_ifstat, 35
- ifs6_out_neighborsolicit

- icmp6_ifstat, 35
- ifs6_out_paramprob
 - icmp6_ifstat, 35
- ifs6_out_pkttoobig
 - icmp6_ifstat, 35
- ifs6_out_redirect
 - icmp6_ifstat, 35
- ifs6_out_request
 - in6_ifstat, 56
- ifs6_out_routeradvert
 - icmp6_ifstat, 35
- ifs6_out_routersolicit
 - icmp6_ifstat, 35
- ifs6_out_timeexceed
 - icmp6_ifstat, 35
- ifs6_reass_fail
 - in6_ifstat, 56
- ifs6_reass_ok
 - in6_ifstat, 56
- ifs6_reass_reqd
 - in6_ifstat, 56
- ifs_bits
 - if_set, 36
- ih_proto
 - ipsec_history, 114
- ih_spi
 - ipsec_history, 114
- im6_dst
 - mrt6msg, 133
 - omrt6msg, 152
- im6_mbz
 - mrt6msg, 133
 - omrt6msg, 152
- im6_mif
 - mrt6msg, 133
 - omrt6msg, 152
- im6_msgtype
 - mrt6msg, 133
 - omrt6msg, 152
- im6_pad
 - mrt6msg, 133
- im6o_multicast_hlim
 - ip6_moptions, 88
- im6o_multicast_ifp
 - ip6_moptions, 88
- im6o_multicast_loop
 - ip6_moptions, 88
- in6.c
 - __P, 249
 - digits, 260
 - faithprefix_p, 260
 - hostnamelen, 249
 - ia62ifa, 249
 - ifa2ia6, 249
 - in6_are_prefix_equal, 249
 - in6_control, 249
 - in6_domifattach, 251
 - in6_domifdetach, 251
 - in6_if2idlen, 252
 - in6_if_up, 252
 - in6_ifaddloop, 252
 - in6_ifawithifp, 252
 - in6_ifinit, 253
 - in6_ifloop_request, 253
 - in6_ifremloop, 253
 - in6_is_addr_deprecated, 254
 - in6_joininggroup, 254
 - in6_leavegroup, 254
 - in6_lifaddr_ioctl, 254
 - in6_localaddr, 255
 - in6_mask2len, 255
 - in6_matchlen, 255
 - in6_multihead, 260
 - in6_prefixlen2mask, 255
 - in6_purgeaddr, 255
 - in6_purgeif, 256
 - in6_setmaxmtu, 256
 - in6_sin6_2_sin, 257
 - in6_sin6_2_sin_in_sock, 257
 - in6_sin_2_v4mapsin6, 257
 - in6_sin_2_v4mapsin6_in_sock, 257
 - in6_unlink_ifa, 257
 - in6_update_ifa, 258
 - in6addr_any, 260
 - in6addr_linklocal_allnodes, 260
 - in6addr_linklocal_allrouters, 260
 - in6addr_loopback, 261
 - in6addr_nodelocal_allnodes, 261
 - in6if_do_dad, 259
 - in6ifa_ifpforlinklocal, 259
 - in6ifa_ifpwithaddr, 259
 - in6mask0, 261
 - in6mask128, 261
 - in6mask32, 261
 - in6mask64, 261
 - in6mask96, 261
 - ip6_sprintf, 259
 - MALLOC_DEFINE, 260
 - MLTMASK_LEN, 249
 - sa6_any, 261
- in6.h
 - __KAME_VERSION, 266
 - __KAME__, 266
 - __P, 280
 - faithprefix_p, 280
 - ICMP6_FILTER, 266
 - IFA6_IS_DEPRECATED, 266
 - IFA6_IS_INVALID, 266

- ifatoia6, 266
- IN6_ARE_ADDR_EQUAL, 266
- IN6_IS_ADDR_LINKLOCAL, 267
- IN6_IS_ADDR_LOOPBACK, 267
- IN6_IS_ADDR_MC_GLOBAL, 267
- IN6_IS_ADDR_MC_INTFACELocal, 267
- IN6_IS_ADDR_MC_LINKLOCAL, 267
- IN6_IS_ADDR_MC_NODELOCAL, 268
- IN6_IS_ADDR_MC_ORGLOCAL, 268
- IN6_IS_ADDR_MC_SITELocal, 268
- IN6_IS_ADDR_MULTICAST, 268
- IN6_IS_ADDR_SITELocal, 268
- IN6_IS_ADDR_UNSPECIFIED, 268
- IN6_IS_ADDR_V4COMPAT, 269
- IN6_IS_ADDR_V4MAPPED, 269
- IN6_IS_SCOPE_LINKLOCAL, 269
- in6addr_any, 280
- in6addr_loopback, 280
- IN6MASK0, 269
- in6mask0, 280
- IN6MASK128, 270
- in6mask128, 280
- IN6MASK32, 270
- in6mask32, 280
- IN6MASK64, 270
- in6mask64, 281
- IN6MASK96, 270
- in6mask96, 281
- INET6_ADDRSTRLEN, 270
- ip6_protoc, 281
- IPV6_2292DSTOPTS, 270
- IPV6_2292HOPLIMIT, 271
- IPV6_2292HOPOPTS, 271
- IPV6_2292NEXTHOP, 271
- IPV6_2292PKTINFO, 271
- IPV6_2292PKTOPTIONS, 271
- IPV6_2292RTHDR, 271
- IPV6_ADDR_INT16_MLL, 271
- IPV6_ADDR_INT16_ULL, 271
- IPV6_ADDR_INT16_USL, 271
- IPV6_ADDR_INT32_MLL, 272
- IPV6_ADDR_INT32_MNL, 272
- IPV6_ADDR_INT32_ONE, 272
- IPV6_ADDR_INT32_SMP, 272
- IPV6_ADDR_INT32_TWO, 272
- IPV6_ADDR_MC_SCOPE, 272
- IPV6_ADDR_SCOPE_GLOBAL, 272
- IPV6_ADDR_SCOPE_INTFACELocal, 272
- IPV6_ADDR_SCOPE_LINKLOCAL, 272
- IPV6_ADDR_SCOPE_NODELOCAL, 273
- IPV6_ADDR_SCOPE_ORGLOCAL, 273
- IPV6_ADDR_SCOPE_SITELocal, 273
- IPV6_AUTOFLOWLABEL, 273
- IPV6_CHECKSUM, 273
- IPV6_DEFAULT_MULTICAST_HOPS, 273
- IPV6_DEFAULT_MULTICAST_LOOP, 273
- IPV6_DONTFRAG, 273
- IPV6_DSTOPTS, 273
- IPV6_FAITH, 274
- IPV6_FW_ADD, 274
- IPV6_FW_DEL, 274
- IPV6_FW_FLUSH, 274
- IPV6_FW_GET, 274
- IPV6_FW_ZERO, 274
- IPV6_HOPLIMIT, 274
- IPV6_HOPOPTS, 274
- IPV6_IPSEC_POLICY, 274
- IPV6_JOIN_GROUP, 275
- IPV6_LEAVE_GROUP, 275
- IPV6_MULTICAST_HOPS, 275
- IPV6_MULTICAST_IF, 275
- IPV6_MULTICAST_LOOP, 275
- IPV6_NEXTHOP, 275
- IPV6_PATHMTU, 275
- IPV6_PKTINFO, 275
- IPV6_PORTRANGE, 276
- IPV6_PORTRANGE_DEFAULT, 276
- IPV6_PORTRANGE_HIGH, 276
- IPV6_PORTRANGE_LOW, 276
- IPV6_PREFER_TEMPADDR, 276
- IPV6_RECVDSTOPTS, 276
- IPV6_RECVHOPLIMIT, 276
- IPV6_RECVHOPOPTS, 276
- IPV6_RECVPATHMTU, 276
- IPV6_RECVPKTINFO, 277
- IPV6_RECVRTHDR, 277
- IPV6_RECVRTHDRDSTOPTS, 277
- IPV6_RECVTCLASS, 277
- IPV6_RTHDR, 277
- IPV6_RTHDR_LOOSE, 277
- IPV6_RTHDR_STRICT, 277
- IPV6_RTHDR_TYPE_0, 277
- IPV6_RTHDRDSTOPTS, 277
- IPV6_SOCKOPT_RESERVED1, 278
- IPV6_TCLASS, 278
- IPV6_UNICAST_HOPS, 278
- IPV6_USE_MIN_MTU, 278
- IPV6_V6ONLY, 278
- M_AUTHIPDGM, 278
- M_AUTHIPHDR, 278
- M_DECRYPTED, 278
- M_LOOP, 278
- s6_addr, 279
- s6_addr16, 279
- s6_addr32, 279
- s6_addr8, 279
- sa6_any, 281

- satosin6, 279
- sin6tosa, 279
- size_t, 279
- socklen_t, 279
- in6_addmulti
 - mld6.c, 470
- in6_addr, 37
 - __u6_addr, 37
 - __u6_addr16, 37
 - __u6_addr32, 37
 - __u6_addr8, 37
- in6_addrlifetime, 38
 - ia6t_expire, 38
 - ia6t_pltime, 38
 - ia6t_preferred, 38
 - ia6t_vltime, 38
- in6_addroute
 - in6_rmx.c, 327
- in6_addrpolicy, 39
 - addr, 39
 - addrmask, 39
 - label, 39
 - preced, 39
 - use, 40
- in6_addrscope
 - scope6.c, 551
- in6_aliasreq, 41
 - ifra_addr, 41
 - ifra_dstaddr, 41
 - ifra_flags, 41
 - ifra_lifetime, 41
 - ifra_name, 42
 - ifra_prefixmask, 42
- IN6_ARE_ADDR_EQUAL
 - in6.h, 266
- IN6_ARE_MASKED_ADDR_EQUAL
 - in6_var.h, 344
- in6_are_prefix_equal
 - in6.c, 249
- IN6_ARE_SCOPE_CMP
 - in6_var.h, 344
- IN6_ARE_SCOPE_EQUAL
 - in6_var.h, 345
- in6_cksum
 - in6_cksum.c, 283
- in6_cksum.c
 - ADDCARRY, 282
 - in6_cksum, 283
 - REDUCE, 282
- in6_clearscope
 - scope6.c, 551
- in6_clsroute
 - in6_rmx.c, 327
- in6_control
 - in6.c, 249
- in6_defrouter, 43
 - expire, 43
 - flags, 43
 - if_index, 43
 - rtaddr, 43
 - rtlifetime, 43
- in6_delmulti
 - mld6.c, 470
- in6_domifattach
 - in6.c, 251
- in6_domifdetach
 - in6.c, 251
- in6_drlist, 45
 - defrouter, 45
 - expire, 45
 - flags, 45
 - if_index, 45
 - ifname, 45
 - rtaddr, 45
 - rtlifetime, 46
- IN6_FIRST_MULTI
 - in6_var.h, 345
- in6_get_hw_ifid
 - in6_ifattach.c, 294
- in6_get_tmpifid
 - in6_ifattach.c, 294
- in6_gif.c
 - gif_encapcheck6, 286
 - gif_validate6, 286
 - in6_gif_attach, 286
 - in6_gif_detach, 286
 - in6_gif_input, 286
 - in6_gif_output, 287
 - in6_gif_protosw, 287
 - inet6domain, 288
- in6_gif.h
 - __P, 289
 - GIF_HLIM, 289
- in6_gif_attach
 - in6_gif.c, 286
- in6_gif_detach
 - in6_gif.c, 286
- in6_gif_input
 - in6_gif.c, 286
- in6_gif_output
 - in6_gif.c, 287
- in6_gif_protosw
 - in6_gif.c, 287
- in6_if2idlen
 - in6.c, 252
- in6_if_up
 - in6.c, 252
- in6_ifadd

- nd6_rtr.c, 526
- in6_ifaddloop
 - in6.c, 252
 - in6_var.h, 355
- in6_ifaddr, 47
 - ia6_createtime, 47
 - ia6_flags, 47
 - ia6_lifetime, 48
 - ia6_ndpr, 48
 - ia6_updatetime, 48
 - ia_addr, 48
 - ia_dstaddr, 48
 - ia_ifa, 48
 - ia_net, 48
 - ia_next, 48
 - ia_plen, 49
 - ia_prefixmask, 49
 - in6_var.h, 357
 - ip6_input.c, 375
- in6_ifattach
 - in6_ifattach.c, 294
- in6_ifattach.c
 - __P, 293
 - EUI64_GBIT, 292
 - EUI64_GROUP, 292
 - EUI64_INDIVIDUAL, 292
 - EUI64_LOCAL, 292
 - EUI64_TO_IFID, 292
 - EUI64_UBIT, 293
 - EUI64_UNIVERSAL, 293
 - generate_tmp_ifid, 293
 - get_ifid, 293
 - get_rand_ifid, 294
 - IFID_LOCAL, 293
 - IFID_UNIVERSAL, 293
 - in6_get_hw_ifid, 294
 - in6_get_tmpifid, 294
 - in6_ifattach, 294
 - in6_ifattach_linklocal, 295
 - in6_ifattach_loopback, 295
 - in6_ifdetach, 296
 - in6_maxmtu, 298
 - in6_nigroup, 297
 - in6_tmpaddrtimer, 297
 - in6_tmpaddrtimer_ch, 298
 - ip6_auto_linklocal, 298
 - ripcbinfo, 298
 - udbinfo, 298
- in6_ifattach.h
 - __P, 299
- in6_ifattach_linklocal
 - in6_ifattach.c, 295
- in6_ifattach_loopback
 - in6_ifattach.c, 295
- IN6_IFAUPDATE_DADDELAY
 - in6_var.h, 345
- in6_ifawithifp
 - in6.c, 252
- in6_ifdetach
 - in6_ifattach.c, 296
- in6_ifextra, 50
 - icmp6_ifstat, 50
 - in6_ifstat, 50
 - nd_ifinfo, 50
 - scope6_id, 50
- IN6_IFF_ANYCAST
 - in6_var.h, 345
- IN6_IFF_AUTOCONF
 - in6_var.h, 345
- IN6_IFF_DEPRECATED
 - in6_var.h, 345
- IN6_IFF_DETACHED
 - in6_var.h, 345
- IN6_IFF_DUPLICATED
 - in6_var.h, 346
- IN6_IFF_NODAD
 - in6_var.h, 346
- IN6_IFF_NOPFX
 - in6_var.h, 346
- IN6_IFF_NOTREADY
 - in6_var.h, 346
- IN6_IFF_TEMPORARY
 - in6_var.h, 346
- IN6_IFF_TENTATIVE
 - in6_var.h, 346
- in6_ifinit
 - in6.c, 253
- in6_ifloop_request
 - in6.c, 253
- in6_ifremloop
 - in6.c, 253
 - in6_var.h, 356
- in6_ifreq, 52
 - ifr_ifru, 52
 - ifr_name, 52
 - ifr_addr, 52
 - ifr_data, 52
 - ifr_dstaddr, 53
 - ifr_flags, 53
 - ifr_flags6, 53
 - ifr_icmp6stat, 53
 - ifr_lifetime, 53
 - ifr_metric, 53
 - ifr_scope_id, 53
 - ifr_stat, 53
- in6_ifstat, 54
 - ifs6_in_addrerr, 54
 - ifs6_in_deliver, 54

- ifs6_in_discard, 54
- ifs6_in_hdrerr, 54
- ifs6_in_mcast, 54
- ifs6_in_noroute, 55
- ifs6_in_protunknown, 55
- ifs6_in_receive, 55
- ifs6_in_toobig, 55
- ifs6_in_truncated, 55
- ifs6_out_discard, 55
- ifs6_out_forward, 55
- ifs6_out_fragcreat, 55
- ifs6_out_fragfail, 55
- ifs6_out_fragok, 55
- ifs6_out_mcast, 55
- ifs6_out_request, 56
- ifs6_reass_fail, 56
- ifs6_reass_ok, 56
- ifs6_reass_reqd, 56
- in6_ifextra, 50
- in6_ifstat_inc
 - in6_var.h, 346
- IN6_IFSTAT_STRICT
 - frag6.c, 228
- in6_init_address_ltimes
 - nd6_rtr.c, 526
- in6_init_prefix_ltimes
 - nd6_rtr.c, 527
- in6_inithead
 - in6_rmx.c, 327
- in6_is_addr_deprecated
 - in6.c, 254
- IN6_IS_ADDR_LINKLOCAL
 - in6.h, 267
- IN6_IS_ADDR_LOOPBACK
 - in6.h, 267
- IN6_IS_ADDR_MC_GLOBAL
 - in6.h, 267
- IN6_IS_ADDR_MC_INTFACELocal
 - in6.h, 267
- IN6_IS_ADDR_MC_LINKLOCAL
 - in6.h, 267
- IN6_IS_ADDR_MC_NODELOCAL
 - in6.h, 268
- IN6_IS_ADDR_MC_ORGLOCAL
 - in6.h, 268
- IN6_IS_ADDR_MC_SITELOCAL
 - in6.h, 268
- IN6_IS_ADDR_MULTICAST
 - in6.h, 268
- IN6_IS_ADDR_SITELOCAL
 - in6.h, 268
- IN6_IS_ADDR_UNSPECIFIED
 - in6.h, 268
- IN6_IS_ADDR_V4COMPAT
 - in6.h, 269
- IN6_IS_ADDR_V4MAPPED
 - in6.h, 269
- IN6_IS_SCOPE_LINKLOCAL
 - in6.h, 269
- in6_joingroup
 - in6.c, 254
 - in6_var.h, 356
- in6_leavegroup
 - in6.c, 254
 - in6_var.h, 356
- in6_lifaddr_ioctl
 - in6.c, 254
- IN6_LINKMTU
 - nd6.h, 498
- in6_localaddr
 - in6.c, 255
- IN6_LOOKUP_MULTI
 - in6_var.h, 347
- in6_losing
 - in6_pcb.c, 302
- in6_mapped_peeraddr
 - in6_pcb.c, 302
- in6_mapped_sockaddr
 - in6_pcb.c, 302
- in6_mask2len
 - in6.c, 255
- in6_matchlen
 - in6.c, 255
- in6_matroute
 - in6_rmx.c, 327
- in6_maxmtu
 - in6_ifattach.c, 298
 - in6_var.h, 357
- in6_mtuxpire
 - in6_rmx.c, 327
- in6_mtutimo
 - in6_rmx.c, 328
- in6_multi, 57
 - in6m_addr, 57
 - in6m_ifma, 57
 - in6m_ifp, 57
 - in6m_refcount, 58
 - in6m_state, 58
 - in6m_timer, 58
 - in6m_timer_ch, 58
 - in6m_timer_expire, 58
 - LIST_ENTRY, 57
- in6_multi_mship, 59
 - i6mm_maddr, 59
 - LIST_ENTRY, 59
- in6_multihead
 - in6.c, 260
- in6_multistep, 60

- i_ia, 60
- i_in6m, 60
- in6_nbrinfo, 61
 - addr, 61
 - asked, 61
 - expire, 61
 - ifname, 61
 - isrouter, 61
 - state, 62
- in6_ndifreq, 63
 - ifindex, 63
 - ifname, 63
- in6_ndireq, 64
 - ifname, 64
 - ndi, 64
- IN6_NEXT_MULTI
 - in6_var.h, 347
- in6_nigroup
 - in6_ifattach.c, 297
- in6_ondireq, 65
 - basereachable, 65
 - chlim, 65
 - flags, 65
 - ifname, 65
 - linkmtu, 65
 - maxmtu, 65
 - ndi, 66
 - reachable, 66
 - recalctm, 66
 - receivedra, 66
 - retrans, 66
- in6_oprlist, 67
 - advrtr, 67
 - advrtrs, 67
 - expire, 67
 - if_index, 67
 - ifname, 68
 - origin, 68
 - ptime, 68
 - prefix, 68
 - prefixlen, 68
 - raflags, 68
 - vtime, 68
- in6_pcb.c
 - in6_losing, 302
 - in6_mapped_peeraddr, 302
 - in6_mapped_sockaddr, 302
 - in6_pcbbind, 303
 - in6_pcbconnect, 303
 - in6_pcbdetach, 304
 - in6_pcbdisconnect, 304
 - in6_pcbfree, 304
 - in6_pcblladdr, 304
 - in6_pcblookup_hash, 305
 - in6_pcblookup_local, 305
 - in6_pcbnotify, 305
 - in6_pcbpurgeif0, 305
 - in6_pcbsetport, 336
 - in6_pktinfo, 69
 - ip6_addr, 69
 - ip6_ifindex, 69
 - in6_prefix, 70
 - advrtrs, 70
 - expire, 70
 - flags, 70
 - if_index, 70
 - origin, 71
 - ptime, 71
 - prefix, 71
 - prefixlen, 71
 - raflags, 71
 - refcnt, 71
 - vtime, 71
- IN6_PREFIX_ND
 - in6_var.h, 347
- in6_pcblookup_local, 305
- in6_pcbnotify, 305
- in6_pcbpurgeif0, 305
- in6_rtchange, 306
- in6_setpeeraddr, 306
- in6_setsockaddr, 306
- in6_sockaddr, 306
- in6_v4mapsin6_sockaddr, 307
- init_sin6, 307
- zeroin6_addr, 307
- in6_pcb.h
 - __P, 309
 - ifatoia6, 308
 - satosin6, 308
 - sin6tosa, 309
- in6_pcbbind
 - in6_pcb.c, 303
- in6_pcbconnect
 - in6_pcb.c, 303
- in6_pcbdetach
 - in6_pcb.c, 304
- in6_pcbdisconnect
 - in6_pcb.c, 304
- in6_pcbfree
 - in6_pcb.c, 304
- in6_pcblladdr
 - in6_pcb.c, 304
- in6_pcblookup_hash
 - in6_pcb.c, 305
- in6_pcblookup_local
 - in6_pcb.c, 305
- in6_pcbnotify
 - in6_pcb.c, 305
- in6_pcbpurgeif0
 - in6_pcb.c, 305
- in6_pcbsetport
 - in6_src.c, 336
- in6_pktinfo, 69
 - ip6_addr, 69
 - ip6_ifindex, 69
- in6_prefix, 70
 - advrtrs, 70
 - expire, 70
 - flags, 70
 - if_index, 70
 - origin, 71
 - ptime, 71
 - prefix, 71
 - prefixlen, 71
 - raflags, 71
 - refcnt, 71
 - vtime, 71
- IN6_PREFIX_ND
 - in6_var.h, 347

- IN6_PREFIX_RR
 - in6_var.h, 347
- in6_prefixlen2mask
 - in6.c, 255
- in6_prefixreq, 72
 - ipr_flags, 72
 - ipr_name, 72
 - ipr_origin, 72
 - ipr_plen, 72
 - ipr_pltime, 72
 - ipr_prefix, 73
 - ipr_vltime, 73
- in6_prflags, 74
 - prf_ra, 74
 - prf_reserved1, 74
 - prf_reserved2, 74
 - prf_reserved3, 74
 - prf_reserved4, 74
 - prf_rr, 75
- in6_prflags::prf_ra, 76
 - autonomous, 76
 - onlink, 76
 - reserved, 76
- in6_prflags::prf_rr, 77
 - decrprefd, 77
 - decrvalid, 77
 - reserved, 77
- in6_prlist, 78
 - advrtr, 78
 - advrtrs, 78
 - expire, 78
 - if_index, 78
 - ifname, 79
 - origin, 79
 - pltime, 79
 - prefix, 79
 - prefixlen, 79
 - raflags, 79
 - vltime, 79
- in6_proto.c
 - __P, 318
 - DOMAIN_SET, 318
 - icmp6_nodeinfo, 319
 - icmp6_rediraccept, 319
 - icmp6_redirtimeout, 320
 - icmp6errppslim, 320
 - inet6domain, 320
 - inet6sw, 320
 - ip6_accept_rtadv, 320
 - ip6_auto_flowlabel, 320
 - ip6_dad_count, 320
 - ip6_defhlim, 321
 - ip6_defmcasthlim, 321
 - ip6_forwarding, 321
 - ip6_gif_hlim, 321
 - ip6_hdrnestlimit, 321
 - ip6_keepfaith, 321
 - ip6_log_interval, 321
 - ip6_log_time, 321
 - ip6_maxfragpackets, 322
 - ip6_maxfrags, 322
 - ip6_mcast_pmtu, 322
 - ip6_rr_prune, 322
 - ip6_sendredirects, 322
 - ip6_use_deprecated, 322
 - ip6_v6only, 322
 - IPV6_SENDREDIRECTS, 315
 - IPV6FORWARDING, 315
 - noursreqs, 322
 - pmtu_expire, 322
 - pmtu_probe, 323
 - PR_ABRTACPTDIS, 315
 - PR_LISTEN, 315
 - rip6_recvspace, 323
 - rip6_sendspace, 323
 - RIPV6RCVQ, 315
 - RIPV6SNDQ, 316
 - SYSCTL_INT, 318
 - sysctl_ip6_temppltime, 318
 - sysctl_ip6_tempvltime, 319
 - SYSCTL_NODE, 319
 - SYSCTL_OID, 319
 - SYSCTL_STRING, 319
 - SYSCTL_STRUCT, 319
 - TUNABLE_INT, 319
 - udp6_recvspace, 323
 - udp6_sendspace, 323
- in6_purgeaddr
 - in6.c, 255
- in6_purgeif
 - in6.c, 256
- in6_rmx.c
 - __P, 327
 - in6_addroute, 327
 - in6_clsroute, 327
 - in6_inithead, 327
 - in6_matroute, 327
 - in6_mtuexpire, 327
 - in6_mtutimo, 328
 - in6_rtqkill, 328
 - in6_rtqtimo, 328
 - MTUTIMO_DEFAULT, 326
 - RTPRF_OURS, 326
 - rtq_minreallyold, 329
 - rtq_mtutimer, 329
 - rtq_reallyold, 329
 - RTQ_TIMEOUT, 326
 - rtq_timeout, 329

- rtq_timer, 329
- rtq_toomany, 329
- SYSCTL_DECL, 328
- SYSCTL_INT, 329
- in6_rrenumreq, 80
 - irr_flags, 80
 - irr_m_len, 80
 - irr_m_maxlen, 80
 - irr_m_minlen, 81
 - irr_matchprefix, 81
 - irr_name, 81
 - irr_origin, 81
 - irr_pltime, 81
 - irr_raflagmask, 81
 - irr_u_keeplen, 81
 - irr_u_uselen, 81
 - irr_useprefix, 81
 - irr_vltime, 81
- in6_rrenumreq::irr_raflagmask, 82
 - autonomous, 82
 - onlink, 82
 - reserved, 82
- in6_rtchange
 - in6_pcb.c, 306
- in6_rtqkill
 - in6_rmx.c, 328
- in6_rtqtimo
 - in6_rmx.c, 328
- in6_selecthlim
 - in6_src.c, 336
- in6_selectif
 - in6_src.c, 336
- in6_selectroute
 - in6_src.c, 336
- in6_selectsrc
 - in6_src.c, 337
- in6_setmaxmtu
 - in6.c, 256
- in6_setpeeraddr
 - in6_pcb.c, 306
- in6_setscope
 - scope6.c, 551
- in6_setsockaddr
 - in6_pcb.c, 306
- in6_sin6_2_sin
 - in6.c, 257
- in6_sin6_2_sin_in_sock
 - in6.c, 257
- in6_sin_2_v4mapsin6
 - in6.c, 257
- in6_sin_2_v4mapsin6_in_sock
 - in6.c, 257
- in6_sockaddr
 - in6_pcb.c, 306
- in6_src.c
 - __P, 335
 - add_addrssel_policyent, 335
 - ADDR_LABEL_NOTAPP, 333
 - ADDRSEL_LOCK, 333
 - addrsel_lock, 339
 - ADDRSEL_LOCK_ASSERT, 333
 - ADDRSEL_LOCK_INIT, 333
 - addrsel_policy_init, 335
 - addrsel_policytab, 339
 - ADDRSEL_SLOCK, 333
 - ADDRSEL_SUNLOCK, 333
 - addrsel_sxlock, 339
 - ADDRSEL_SXLOCK_INIT, 333
 - ADDRSEL_UNLOCK, 334
 - ADDRSEL_XLOCK, 334
 - ADDRSEL_XUNLOCK, 334
 - BREAK, 334
 - defaultaddrpolicy, 339
 - delete_addrssel_policyent, 335
 - dump_addrssel_policyent, 336
 - in6_pcbsetport, 336
 - in6_selecthlim, 336
 - in6_selectif, 336
 - in6_selectroute, 336
 - in6_selectsrc, 337
 - in6_src_ioctl, 337
 - in6_src_sysctl, 337
 - init_policy_queue, 338
 - ip6_prefer_tempaddr, 339
 - lookup_addrssel_policy, 338
 - match_addrssel_policy, 338
 - NEXT, 334
 - REPLACE, 334
 - selectroute, 338
 - SYSCTL_DECL, 339
 - SYSCTL_NODE, 339
 - TAILQ_HEAD, 339
 - walk_addrssel_policy, 339
- in6_src_ioctl
 - in6_src.c, 337
- in6_src_sysctl
 - in6_src.c, 337
- in6_tmpaddrtimer
 - in6_ifattach.c, 297
- in6_tmpaddrtimer_ch
 - in6_ifattach.c, 298
 - ip6_input.c, 375
 - nd6.c, 492
- in6_tmpifadd
 - nd6_rtr.c, 527
- in6_unlink_ifa
 - in6.c, 257
- in6_update_ifa

- in6.c, 258
- in6_v4mapsin6_sockaddr
 - in6_pcb.c, 307
- in6_var.h
 - __P, 355
 - IA6_DSTIN6, 343
 - IA6_DSTSIN6, 343
 - IA6_IN6, 343
 - IA6_MASKIN6, 343
 - IA6_SIN6, 343
 - ia_flags, 343
 - ia_ifp, 344
 - icmp6stat, 356
 - IFA_DSTIN6, 344
 - IFA_IN6, 344
 - IFP_TO_IA6, 344
 - IFPR_IN6, 344
 - IN6_ARE_MASKED_ADDR_EQUAL, 344
 - IN6_ARE_SCOPE_CMP, 344
 - IN6_ARE_SCOPE_EQUAL, 345
 - IN6_FIRST_MULTI, 345
 - in6_ifaddloop, 355
 - in6_ifaddr, 357
 - IN6_IFAUPDATE_DADDELAY, 345
 - IN6_IFF_ANYCAST, 345
 - IN6_IFF_AUTOCONF, 345
 - IN6_IFF_DEPRECATED, 345
 - IN6_IFF_DETACHED, 345
 - IN6_IFF_DUPLICATED, 346
 - IN6_IFF_NODAD, 346
 - IN6_IFF_NOPFX, 346
 - IN6_IFF_NOTREADY, 346
 - IN6_IFF_TEMPORARY, 346
 - IN6_IFF_TENTATIVE, 346
 - in6_ifremloop, 356
 - in6_ifstat_inc, 346
 - in6_joingroup, 356
 - in6_leavegroup, 356
 - IN6_LOOKUP_MULTI, 347
 - in6_maxmtu, 357
 - IN6_NEXT_MULTI, 347
 - IN6_PREFIX_ND, 347
 - IN6_PREFIX_RR, 347
 - IN6M_TIMER_UNDEF, 347
 - inet6ctlerrmap, 357
 - ipr_raf_auto, 347
 - ipr_raf_onlink, 348
 - ipr_rrf_decrprefd, 348
 - ipr_rrf_decrvalid, 348
 - ipr_statef_onlink, 348
 - irr_raf_auto, 348
 - irr_raf_mask_auto, 348
 - irr_raf_mask_onlink, 348
 - irr_raf_mask_reserved, 348
 - irr_raf_onlink, 348
 - irr_rrf, 348
 - irr_rrf_decrprefd, 348
 - irr_rrf_decrvalid, 349
 - irr_statef_onlink, 349
 - LIST_HEAD, 356
 - OSIOCGIFINFO_IN6, 349
 - PR_ORIG_KERNEL, 349
 - PR_ORIG_RA, 349
 - PR_ORIG_RR, 349
 - PR_ORIG_STATIC, 349
 - SIOCAADDRCTL_POLICY, 349
 - SIOCAIFADDR_IN6, 349
 - SIOCAIFPREFIX_IN6, 349
 - SIOCCIFPREFIX_IN6, 350
 - SIOCADDRCTL_POLICY, 350
 - SIOCDEFADDR_IN6, 350
 - SIOCDEFPREFIX_IN6, 350
 - SIOCGDEFIFACE_IN6, 350
 - SIOCGDRLST_IN6, 350
 - SIOCGETMIFCNT_IN6, 350
 - SIOCGETSGCNT_IN6, 351
 - SIOCGIFADDR_IN6, 351
 - SIOCGIFAFLAG_IN6, 351
 - SIOCGIFALIFETIME_IN6, 351
 - SIOCGIFDSTADDR_IN6, 351
 - SIOCGIFINFO_IN6, 351
 - SIOCGIFNETMASK_IN6, 351
 - SIOCGIFPDSTADDR_IN6, 351
 - SIOCGIFPREFIX_IN6, 352
 - SIOCGIFPSRCADDR_IN6, 352
 - SIOCGIFSTAT_ICMP6, 352
 - SIOCGIFSTAT_IN6, 352
 - SIOCGNBRINFO_IN6, 352
 - SIOCGPRLST_IN6, 352
 - SIOCGSCOPE6, 352
 - SIOCGSCOPE6DEF, 352
 - SIOCSDEFIFACE_IN6, 352
 - SIOCSGIFPREFIX_IN6, 353
 - SIOCSIFADDR_IN6, 353
 - SIOCSIFALIFETIME_IN6, 353
 - SIOCSIFDSTADDR_IN6, 353
 - SIOCSIFINFO_FLAGS, 353
 - SIOCSIFINFO_IN6, 353
 - SIOCSIFNETMASK_IN6, 353
 - SIOCSIFPHYADDR_IN6, 353
 - SIOCSIFPREFIX_IN6, 354
 - SIOCSNDFLUSH_IN6, 354
 - SIOCSPFXFLUSH_IN6, 354
 - SIOCSRTRFLUSH_IN6, 354
 - SIOCSSCOPE6, 354
 - zeroin6_addr, 357
- in6addr_any
 - in6.c, 260

- in6.h, 280
- in6addr_linklocal_allnodes
 - in6.c, 260
- in6addr_linklocal_allrouters
 - in6.c, 260
- in6addr_loopback
 - in6.c, 261
 - in6.h, 280
- in6addr_nodelocal_allnodes
 - in6.c, 261
- in6if_do_dad
 - in6.c, 259
- in6ifa_ifpforlinklocal
 - in6.c, 259
- in6ifa_ifpwithaddr
 - in6.c, 259
- in6m_addr
 - in6_multi, 57
- in6m_ifma
 - in6_multi, 57
- in6m_ifp
 - in6_multi, 57
- in6m_refcount
 - in6_multi, 58
- in6m_state
 - in6_multi, 58
- in6m_timer
 - in6_multi, 58
- in6m_timer_ch
 - in6_multi, 58
- in6m_timer_expire
 - in6_multi, 58
- IN6M_TIMER_UNDEF
 - in6_var.h, 347
- IN6MASK0
 - in6.h, 269
- in6mask0
 - in6.c, 261
 - in6.h, 280
- IN6MASK128
 - in6.h, 270
- in6mask128
 - in6.c, 261
 - in6.h, 280
- IN6MASK32
 - in6.h, 270
- in6mask32
 - in6.c, 261
 - in6.h, 280
- IN6MASK64
 - in6.h, 270
- in6mask64
 - in6.c, 261
 - in6.h, 281
- IN6MASK96
 - in6.h, 270
- in6mask96
 - in6.c, 261
 - in6.h, 281
- in6pcb
 - udp6_output.c, 567
- in_ahauthfail
 - ipsecstat, 119
- in_ahauthsucc
 - ipsecstat, 119
- in_ahhist
 - ipsecstat, 119
- in_ahreplay
 - ipsecstat, 120
- in_badspi
 - ipsecstat, 120
- in_comphist
 - ipsecstat, 120
- in_espathfail
 - ipsecstat, 120
- in_espathsucc
 - ipsecstat, 120
- in_esphist
 - ipsecstat, 120
- in_espreplay
 - ipsecstat, 120
- in_inval
 - ipsecstat, 120
- in_nomem
 - ipsecstat, 120
- in_nosa
 - ipsecstat, 120
- in_polvio
 - ipsecstat, 120
- in_success
 - ipsecstat, 121
- INET6_ADDRSTRLEN
 - in6.h, 270
- inet6_ndpr_msghdr, 83
- inpm_msglen, 83
- inpm_prefix, 83
- inpm_type, 83
- inpm_version, 83
- prm_expire, 84
- prm_flags, 84
- prm_index, 84
- prm_plen, 84
- prm_pltime, 84
- prm_preferred, 84
- prm_vltim, 84
- inet6_pfil_hook
 - ip6_input.c, 375
 - ip6_var.h, 418

- inet6ctlerrmap
 - in6_var.h, 357
 - ip6_input.c, 375
- inet6domain
 - icmp6.c, 243
 - in6_gif.c, 288
 - in6_proto.c, 320
 - ip6_input.c, 376
- inet6sw
 - in6_proto.c, 320
 - ip6protosw.h, 423
- inetsw
 - sctp6_usrreq.c, 562
 - udp6_usrreq.c, 576
- init_policy_queue
 - in6_src.c, 338
- init_sin6
 - in6_pcb.c, 307
- initialized
 - nd_ifinfo, 140
- initid
 - ip6_id.c, 364
- inpcbpolicy, 85
 - cache, 85
 - cacheflags, 85
 - cachegen, 85
 - cacheidx, 85
 - priv, 86
 - sp_in, 86
 - sp_out, 86
- inpm_msglen
 - inet6_ndpr_msghdr, 83
- inpm_prefix
 - inet6_ndpr_msghdr, 83
- inpm_type
 - inet6_ndpr_msghdr, 83
- inpm_version
 - inet6_ndpr_msghdr, 83
- installed
 - nd_defrouter, 139
- INT32_MAX
 - ip6_id.c, 364
- ip4_ah_clearatos
 - ipsec.c, 453
- ip4_ah_net_deflev
 - ipsec.c, 453
- ip4_ah_offsetmask
 - ipsec.c, 454
- ip4_ah_trans_deflev
 - ipsec.c, 454
- ip4_def_policy
 - ipsec.c, 454
- ip4_esp_net_deflev
 - ipsec.c, 454
- ip4_esp_randpad
 - ipsec.c, 454
- ip4_esp_trans_deflev
 - ipsec.c, 454
- ip4_ipsec_dfbit
 - ipsec.c, 454
- ip4_ipsec_ecn
 - ipsec.c, 454
- ip6_accept_rtadv
 - in6_proto.c, 320
 - ip6_var.h, 419
- ip6_addaux
 - ip6_input.c, 370
- ip6_ah_net_deflev
 - ipsec6.h, 466
- ip6_ah_trans_deflev
 - ipsec6.h, 466
- ip6_anonportmax
 - ip6_var.h, 419
- ip6_anonportmin
 - ip6_var.h, 419
- ip6_auto_flowlabel
 - in6_proto.c, 320
 - ip6_var.h, 419
- ip6_auto_linklocal
 - in6_ifattach.c, 298
 - ip6_var.h, 419
- ip6_clearpktopts
 - ip6_output.c, 402
- ip6_copyexthdr
 - ip6_output.c, 402
- ip6_copypktopts
 - ip6_output.c, 402
- ip6_ctloutput
 - ip6_output.c, 403
- ip6_dad_count
 - in6_proto.c, 320
 - ip6_var.h, 419
- ip6_def_policy
 - ipsec6.h, 466
- ip6_defhlim
 - in6_proto.c, 321
 - ip6_var.h, 419
- ip6_defmcasthlim
 - in6_proto.c, 321
 - ip6_var.h, 419
- ip6_delaux
 - ip6_input.c, 370
- ip6_desync_factor
 - nd6.h, 506
 - nd6_rtr.c, 534
- ip6_ecn.h
 - ip6_ecn_egress, 359
 - ip6_ecn_ingress, 359

- ip6_ecn_egress
 - ip6_ecn.h, 359
- ip6_ecn_ingress
 - ip6_ecn.h, 359
- ip6_esp_net_deflev
 - ipsec6.h, 466
- ip6_esp_randpad
 - ipsec6.h, 466
- ip6_esp_trans_deflev
 - ipsec6.h, 466
- ip6_exthdrs, 87
 - ip6e_dest1, 87
 - ip6e_dest2, 87
 - ip6e_hbh, 87
 - ip6e_ip6, 87
 - ip6e_rthdr, 87
- ip6_findaux
 - ip6_input.c, 370
- ip6_forward
 - ip6_forward.c, 362
- ip6_forward.c
 - ip6_forward, 362
 - ip6_forward_rt, 362
- ip6_forward_rt
 - ip6_forward.c, 362
 - ip6_input.c, 376
- ip6_forward_srcrt
 - ip6_input.c, 376
 - ip6_var.h, 419
- ip6_forwarding
 - in6_proto.c, 321
 - ip6_var.h, 419
- ip6_freemoptions
 - ip6_output.c, 403
- ip6_freepcbopts
 - ip6_output.c, 404
- ip6_get_prevhdr
 - ip6_input.c, 370
- ip6_getdstifaddr
 - ip6_input.c, 370
- ip6_getmoptions
 - ip6_output.c, 404
- ip6_getpcbopt
 - ip6_output.c, 404
- ip6_getpmtu
 - ip6_output.c, 404
- ip6_gif_hlim
 - in6_proto.c, 321
 - ip6_var.h, 420
- IP6_HDR_ALIGNED_P
 - ip6_var.h, 413
- ip6_hdrnestlimit
 - in6_proto.c, 321
 - ip6_var.h, 420
- ip6_hopopts_input
 - ip6_input.c, 371
- ip6_id.c
 - initid, 364
 - INT32_MAX, 364
 - ip6_randomflowlabel, 364
 - ip6_randomid, 364
 - pmod, 365
 - randomid, 365
 - randomtab_20, 365
 - randomtab_32, 365
- ip6_init
 - ip6_input.c, 371
- ip6_init2
 - ip6_input.c, 371
- ip6_initpktopts
 - ip6_output.c, 405
- ip6_input
 - ip6_input.c, 372
- ip6_input.c
 - __P, 370
 - in6_ifaddr, 375
 - in6_tmpaddrtimer_ch, 375
 - inet6_pfil_hook, 375
 - inet6ctlerrmap, 375
 - inet6domain, 376
 - ip6_addaux, 370
 - ip6_delaux, 370
 - ip6_findaux, 370
 - ip6_forward_rt, 376
 - ip6_forward_srcrt, 376
 - ip6_get_prevhdr, 370
 - ip6_getdstifaddr, 370
 - ip6_hopopts_input, 371
 - ip6_init, 371
 - ip6_init2, 371
 - ip6_input, 372
 - ip6_lasthdr, 373
 - ip6_nexthdr, 373
 - ip6_notify_pmtu, 373
 - ip6_ours_check_algorithm, 376
 - ip6_process_hopopts, 373
 - ip6_protox, 376
 - ip6_savecontrol, 374
 - ip6_setdstifaddr, 374
 - ip6_sourcecheck, 376
 - ip6_sourcecheck_interval, 376
 - ip6_unknown_opt, 374
 - ip6intrlq, 376
 - ip6qmaxlen, 376
 - ip6stat, 376
 - IS2292, 370
 - M2MMAX, 370
 - rt6_key, 370

- SYSINIT, 375
- ip6_insert_jumboopt
 - ip6_output.c, 405
- ip6_insertfraghdr
 - ip6_output.c, 405
- ip6_ipsec_ecn
 - ipsec6.h, 466
- ip6_keepfaith
 - in6_proto.c, 321
 - ip6_var.h, 420
- ip6_lasthdr
 - ip6_input.c, 373
- ip6_log_interval
 - in6_proto.c, 321
 - ip6_var.h, 420
- ip6_log_time
 - in6_proto.c, 321
 - ip6_var.h, 420
- ip6_lowportmax
 - ip6_var.h, 420
- ip6_lowportmin
 - ip6_var.h, 420
- ip6_maxfragpackets
 - in6_proto.c, 322
 - ip6_var.h, 420
- ip6_maxfrags
 - in6_proto.c, 322
 - ip6_var.h, 420
- ip6_mcast_pmtu
 - in6_proto.c, 322
 - ip6_var.h, 420
- ip6_mdq
 - ip6_mrout.c, 385
- ip6_mforward
 - ip6_mrout.c, 386
- ip6_mloopback
 - ip6_output.c, 405
- ip6_moptions, 88
 - im6o_multicast_hlim, 88
 - im6o_multicast_ifp, 88
 - im6o_multicast_loop, 88
 - LIST_HEAD, 88
- ip6_mrout.c
 - __P, 384
 - add_m6fc, 384
 - add_m6if, 384
 - del_m6fc, 384
 - del_m6if, 385
 - ENCAP_HOPS, 381
 - EXPIRE_TIMEOUT, 381
 - expire_upcalls, 385
 - expire_upcalls_ch, 390
 - get_mif6_cnt, 385
 - get_sg_cnt, 385
 - ip6_mdq, 385
 - ip6_mforward, 386
 - ip6_mrout, 390
 - ip6_mrout_done, 386
 - ip6_mrout_get, 386
 - ip6_mrout_init, 387
 - ip6_mrout_set, 387
 - ip6_mrout_ver, 390
 - ip6_mrtproto, 390
 - M_HASCL, 381
 - MALLOC_DEFINE, 387
 - MC6_SEND, 381
 - MF6CFIND, 381
 - MF6CHASH, 382
 - mf6ctable, 390
 - mif6table, 390
 - mrt6_ioctl, 387
 - mrt6stat, 390
 - multicast_register_if6, 390
 - n6expire, 390
 - NO_RTE_FOUND, 382
 - nummifs, 390
 - phyint_send, 388
 - pim6, 390
 - PIM6_CHECKSUM, 382
 - pim6_input, 388
 - pim6stat, 391
 - reg_mif_num, 391
 - register_send, 389
 - RTE_FOUND, 382
 - set_pim6, 389
 - sin6, 391
 - socket_send, 389
 - TV_DELTA, 382
 - TV_LT, 383
 - UPCALL_EXPIRE, 383
- ip6_mrout.h
 - __P, 396
 - GET_TIME, 393
 - howmany, 393
 - IF_CLR, 393
 - IF_COPY, 393
 - IF_ISSET, 393
 - if_mask, 396
 - IF_SET, 393
 - IF_SETSIZE, 393
 - IF_ZERO, 393
 - MAX_UPQ6, 394
 - MAXMIFS, 394
 - MF6C_INCOMPLETE_PARENT, 394
 - MF6CHASHMOD, 394
 - MF6CTBSIZ, 394
 - MIFF_REGISTER, 394
 - mifi_t, 396

- MRT6_ADD_MFC, 394
- MRT6_ADD_MIF, 394
- MRT6_DEL_MFC, 394
- MRT6_DEL_MIF, 395
- MRT6_DONE, 395
- MRT6_INIT, 395
- MRT6_OINIT, 395
- MRT6_PIM, 395
- MRT6MSG_NOCACHE, 395
- MRT6MSG_WHOLEPKT, 395
- MRT6MSG_WRONGMIF, 395
- NIFBITS, 395
- ip6_mrouter
 - ip6_mroute.c, 390
 - ip6_var.h, 421
- ip6_mrouter_done
 - ip6_mroute.c, 386
- ip6_mrouter_get
 - ip6_mroute.c, 386
- ip6_mrouter_init
 - ip6_mroute.c, 387
- ip6_mrouter_set
 - ip6_mroute.c, 387
- ip6_mrouter_ver
 - ip6_mroute.c, 390
- ip6_mrtproto
 - ip6_mroute.c, 390
- ip6_mtuinfo, 89
 - ip6m_addr, 89
 - ip6m_mtu, 89
- ip6_nexthdr
 - ip6_input.c, 373
- ip6_notify_pmtu
 - ip6_input.c, 373
- ip6_optlen
 - ip6_output.c, 405
- ip6_opts
 - mld6.c, 474
- ip6_ours_check_algorithm
 - ip6_input.c, 376
- ip6_output
 - ip6_output.c, 405
- ip6_output.c
 - __P, 402
 - copypktopts, 402
 - elen, 400
 - ip6_clearpktopts, 402
 - ip6_copyexthdr, 402
 - ip6_copypktopts, 402
 - ip6_ctloutput, 403
 - ip6_freemoptions, 403
 - ip6_freepcbopts, 404
 - ip6_getmoptions, 404
 - ip6_getpcbopt, 404
 - ip6_getpmtu, 404
 - ip6_initpktopts, 405
 - ip6_insert_jumboopt, 405
 - ip6_insertfraghdr, 405
 - ip6_mloopback, 405
 - ip6_optlen, 405
 - ip6_output, 405
 - ip6_pcbopt, 407
 - ip6_pcbopts, 407
 - ip6_raw_ctloutput, 408
 - ip6_setmoptions, 408
 - ip6_setpktopt, 409
 - ip6_setpktopts, 409
 - ip6_splithdr, 409
 - JUMBOOPTLEN, 400
 - MAKE_CHAIN, 400
 - MAKE_EXTHDR, 400
 - MALLOC_DEFINE, 409
 - OPTBIT, 400
 - OPTSET, 400
 - OPTSET2292, 401
 - PKTOPT_EXTHDRCPY, 401
- ip6_pcbopt
 - ip6_output.c, 407
- ip6_pcbopts
 - ip6_output.c, 407
- ip6_pktopts, 90
 - ip6po_dest1, 90
 - ip6po_dest2, 90
 - ip6po_flags, 90
 - ip6po_hbh, 91
 - ip6po_hlim, 91
 - ip6po_m, 91
 - ip6po_minmtu, 91
 - ip6po_nhinfo, 91
 - ip6po_pktinfo, 91
 - ip6po_prefer_tempaddr, 91
 - ip6po_rhinfo, 91
 - ip6po_tclass, 91
- ip6_prefer_tempaddr
 - in6_src.c, 339
 - ip6_var.h, 421
- ip6_process_hopopts
 - ip6_input.c, 373
- ip6_protoc
 - in6.h, 281
 - ip6_input.c, 376
- ip6_randomflowlabel
 - ip6_id.c, 364
- ip6_randomid
 - ip6_id.c, 364
- ip6_raw_ctloutput
 - ip6_output.c, 408
- IP6_REASS_MBUF

- ip6_var.h, 413
- ip6_rr_prune
 - in6_proto.c, 322
 - ip6_var.h, 421
- ip6_rthdr0
 - route6.c, 547
- ip6_savecontrol
 - ip6_input.c, 374
- ip6_sendredirects
 - in6_proto.c, 322
 - ip6_var.h, 421
- ip6_setdstifaddr
 - ip6_input.c, 374
- ip6_setmoptions
 - ip6_output.c, 408
- ip6_setpktopt
 - ip6_output.c, 409
- ip6_setpktopts
 - ip6_output.c, 409
- ip6_sourcecheck
 - ip6_input.c, 376
 - ip6_var.h, 421
- ip6_sourcecheck_interval
 - ip6_input.c, 376
 - ip6_var.h, 421
- ip6_splithdr
 - ip6_output.c, 409
- ip6_sprintf
 - in6.c, 259
- ip6_temp_preferred_lifetime
 - nd6.h, 507
 - nd6_rtr.c, 534
- ip6_temp_regen_advance
 - nd6.h, 507
 - nd6_rtr.c, 534
- ip6_temp_valid_lifetime
 - nd6.h, 507
 - nd6_rtr.c, 534
- ip6_unknown_opt
 - ip6_input.c, 374
- ip6_use_defzone
 - ip6_var.h, 421
 - scope6.c, 553
- ip6_use_deprecated
 - in6_proto.c, 322
 - ip6_var.h, 421
- ip6_use_tempaddr
 - ip6_var.h, 421
 - nd6_rtr.c, 534
- ip6_v6only
 - in6_proto.c, 322
 - ip6_var.h, 422
- ip6_var.h
 - __P, 418
- inet6_pfil_hook, 418
- ip6_accept_rtadv, 419
- ip6_anonportmax, 419
- ip6_anonportmin, 419
- ip6_auto_flowlabel, 419
- ip6_auto_linklocal, 419
- ip6_dad_count, 419
- ip6_defhlim, 419
- ip6_defmcasthlim, 419
- ip6_forward_srcrt, 419
- ip6_forwarding, 419
- ip6_gif_hlim, 420
- IP6_HDR_ALIGNED_P, 413
- ip6_hdrnestlimit, 420
- ip6_keepfaith, 420
- ip6_log_interval, 420
- ip6_log_time, 420
- ip6_lowportmax, 420
- ip6_lowportmin, 420
- ip6_maxfragpackets, 420
- ip6_maxfrags, 420
- ip6_mcast_pmtu, 420
- ip6_mrouter, 421
- ip6_prefer_tempaddr, 421
- IP6_REASS_MBUF, 413
- ip6_rr_prune, 421
- ip6_sendredirects, 421
- ip6_sourcecheck, 421
- ip6_sourcecheck_interval, 421
- ip6_use_defzone, 421
- ip6_use_deprecated, 421
- ip6_use_tempaddr, 421
- ip6_v6only, 422
- IP6A_BRUID, 414
- IP6A_HASEEN, 414
- IP6A_RTALERTSEEN, 414
- IP6A_SWAP, 414
- IP6PO_DONTFRAG, 414
- IP6PO_MINMTU_ALL, 414
- IP6PO_MINMTU_DISABLE, 414
- IP6PO_MINMTU_MCASTONLY, 414
- ip6po_nexthop, 414
- ip6po_nextroute, 414
- ip6po_route, 415
- ip6po_rthdr, 415
- IP6PO_TEMPADDR_NOTPREFER, 415
- IP6PO_TEMPADDR_PREFER, 415
- IP6PO_TEMPADDR_SYSTEM, 415
- IP6PO_USECOA, 415
- ip6stat, 422
- IPV6_FORWARDING, 415
- IPV6_MINMTU, 415
- IPV6_UNSPECSRC, 415
- rip6_usrreqs, 422

- IP6A_BRUID
 - ip6_var.h, 414
- ip6a_bruid
 - ip6aux, 95
- ip6a_careof
 - ip6aux, 95
- ip6a_dstia6
 - ip6aux, 95
- ip6a_flags
 - ip6aux, 95
- IP6A_HASEEN
 - ip6_var.h, 414
- ip6a_home
 - ip6aux, 96
- ip6a_rtalert
 - ip6aux, 96
- IP6A_RTALERTSEEN
 - ip6_var.h, 414
- IP6A_SWAP
 - ip6_var.h, 414
- ip6af_down
 - ip6asfrag, 93
- ip6af_frglen
 - ip6asfrag, 93
- ip6af_head
 - ip6asfrag, 93
- ip6af_hlim
 - ip6asfrag, 93
- ip6af_len
 - ip6asfrag, 93
- ip6af_m
 - ip6asfrag, 94
- ip6af_mff
 - ip6asfrag, 94
- ip6af_nxt
 - ip6asfrag, 94
- ip6af_off
 - ip6asfrag, 94
- ip6af_offset
 - ip6asfrag, 94
- ip6af_up
 - ip6asfrag, 94
- ip6asfrag, 93
 - ip6af_down, 93
 - ip6af_frglen, 93
 - ip6af_head, 93
 - ip6af_hlim, 93
 - ip6af_len, 93
 - ip6af_m, 94
 - ip6af_mff, 94
 - ip6af_nxt, 94
 - ip6af_off, 94
 - ip6af_offset, 94
 - ip6af_up, 94
- ip6aux, 95
 - ip6a_bruid, 95
 - ip6a_careof, 95
 - ip6a_dstia6, 95
 - ip6a_flags, 95
 - ip6a_home, 96
 - ip6a_rtalert, 96
- ip6c_cmdarg
 - ip6ctlparam, 97
- ip6c_dst
 - ip6ctlparam, 97
- ip6c_finaldst
 - ip6ctlparam, 97
- ip6c_icmp6
 - ip6ctlparam, 97
- ip6c_ip6
 - ip6ctlparam, 98
- ip6c_m
 - ip6ctlparam, 98
- ip6c_nxt
 - ip6ctlparam, 98
- ip6c_off
 - ip6ctlparam, 98
- ip6c_src
 - ip6ctlparam, 98
- ip6ctlparam, 97
 - ip6c_cmdarg, 97
 - ip6c_dst, 97
 - ip6c_finaldst, 97
 - ip6c_icmp6, 97
 - ip6c_ip6, 98
 - ip6c_m, 98
 - ip6c_nxt, 98
 - ip6c_off, 98
 - ip6c_src, 98
- ip6e_dest1
 - ip6_exthdrs, 87
- ip6e_dest2
 - ip6_exthdrs, 87
- ip6e_hbh
 - ip6_exthdrs, 87
- ip6e_ip6
 - ip6_exthdrs, 87
- ip6e_rthdr
 - ip6_exthdrs, 87
- ip6intrq
 - ip6_input.c, 376
- ip6m_addr
 - ip6_mtuintf, 89
- ip6m_mtu
 - ip6_mtuintf, 89
- ip6po_dest1
 - ip6_pktopts, 90
- ip6po_dest2

- ip6_pktopts, 90
- IP6PO_DONTFRAG
 - ip6_var.h, 414
- ip6po_flags
 - ip6_pktopts, 90
- ip6po_hbh
 - ip6_pktopts, 91
- ip6po_hlim
 - ip6_pktopts, 91
- ip6po_m
 - ip6_pktopts, 91
- ip6po_minmtu
 - ip6_pktopts, 91
- IP6PO_MINMTU_ALL
 - ip6_var.h, 414
- IP6PO_MINMTU_DISABLE
 - ip6_var.h, 414
- IP6PO_MINMTU_MCASTONLY
 - ip6_var.h, 414
- ip6po_nexthop
 - ip6_var.h, 414
- ip6po_nextroute
 - ip6_var.h, 414
- ip6po_nhi_nexthop
 - ip6po_nhinfo, 99
- ip6po_nhi_route
 - ip6po_nhinfo, 99
- ip6po_nhinfo, 99
 - ip6_pktopts, 91
 - ip6po_nhi_nexthop, 99
 - ip6po_nhi_route, 99
- ip6po_pktinfo
 - ip6_pktopts, 91
- ip6po_prefer_tempaddr
 - ip6_pktopts, 91
- ip6po_rhi_route
 - ip6po_rhinfo, 100
- ip6po_rhi_rthdr
 - ip6po_rhinfo, 100
- ip6po_rhinfo, 100
 - ip6_pktopts, 91
 - ip6po_rhi_route, 100
 - ip6po_rhi_rthdr, 100
- ip6po_route
 - ip6_var.h, 415
- ip6po_rthdr
 - ip6_var.h, 415
- ip6po_tclass
 - ip6_pktopts, 91
- IP6PO_TEMPADDR_NOTPREFER
 - ip6_var.h, 415
- IP6PO_TEMPADDR_PREFER
 - ip6_var.h, 415
- IP6PO_TEMPADDR_SYSTEM
 - ip6_var.h, 415
- IP6PO_USECOA
 - ip6_var.h, 415
- ip6protosw, 101
 - __P, 102
 - pr_domain, 102
 - pr_flags, 102
 - pr_protocol, 102
 - pr_type, 102
 - pr_usrreqs, 102
- ip6protosw.h
 - inet6sw, 423
- ip6q, 103
 - frag6.c, 232
 - ip6q_arrive, 103
 - ip6q_down, 103
 - ip6q_dst, 103
 - ip6q_head, 104
 - ip6q_hlim, 104
 - ip6q_ident, 104
 - ip6q_len, 104
 - ip6q_next, 104
 - ip6q_nfrag, 104
 - ip6q_nxt, 104
 - ip6q_prev, 104
 - ip6q_ttl, 104
 - ip6q_unfrglen, 104
 - ip6q_up, 105
- ip6q_arrive
 - ip6q, 103
- ip6q_down
 - ip6q, 103
- ip6q_dst
 - ip6q, 103
- ip6q_head
 - ip6q, 104
- ip6q_hlim
 - ip6q, 104
- ip6q_ident
 - ip6q, 104
- ip6q_len
 - ip6q, 104
- IP6Q_LOCK
 - frag6.c, 228
- IP6Q_LOCK_ASSERT
 - frag6.c, 228
- IP6Q_LOCK_INIT
 - frag6.c, 228
- ip6q_next
 - ip6q, 104
- ip6q_nfrag
 - ip6q, 104
- ip6q_nxt
 - ip6q, 104

- ip6q_prev
 - ip6q, 104
- IP6Q_TRYLOCK
 - frag6.c, 228
- ip6q_ttl
 - ip6q, 104
- ip6q_unfrglen
 - ip6q, 104
- IP6Q_UNLOCK
 - frag6.c, 228
- ip6q_up
 - ip6q, 105
- ip6qlock
 - frag6.c, 232
- ip6qmaxlen
 - ip6_input.c, 376
- ip6s_badoptions
 - ip6stat, 107
- ip6s_badscope
 - ip6stat, 107
- ip6s_badvers
 - ip6stat, 107
- ip6s_cantforward
 - ip6stat, 107
- ip6s_cantfrag
 - ip6stat, 107
- ip6s_delivered
 - ip6stat, 107
- ip6s_exthdrtoolong
 - ip6stat, 107
- ip6s_forward
 - ip6stat, 107
- ip6s_forward_cachehit
 - ip6stat, 107
- ip6s_forward_cachemiss
 - ip6stat, 107
- ip6s_fragdropped
 - ip6stat, 108
- ip6s_fragmented
 - ip6stat, 108
- ip6s_fragments
 - ip6stat, 108
- ip6s_fragoverflow
 - ip6stat, 108
- ip6s_fragtimeout
 - ip6stat, 108
- ip6s_localout
 - ip6stat, 108
- ip6s_m1
 - ip6stat, 108
- ip6s_m2m
 - ip6stat, 108
- ip6s_mext1
 - ip6stat, 108
- ip6s_mext2m
 - ip6stat, 108
- ip6s_nogif
 - ip6stat, 109
- ip6s_noroute
 - ip6stat, 109
- ip6s_notmember
 - ip6stat, 109
- ip6s_nxthist
 - ip6stat, 109
- ip6s_odropped
 - ip6stat, 109
- ip6s_ofragments
 - ip6stat, 109
- ip6s_rawout
 - ip6stat, 109
- ip6s_reassembled
 - ip6stat, 109
- ip6s_redirectsent
 - ip6stat, 109
- ip6s_sources_deprecated
 - ip6stat, 109
- ip6s_sources_none
 - ip6stat, 110
- ip6s_sources_otherif
 - ip6stat, 110
- ip6s_sources_otherscope
 - ip6stat, 110
- ip6s_sources_rule
 - ip6stat, 110
- ip6s_sources_sameif
 - ip6stat, 110
- ip6s_sources_samescope
 - ip6stat, 110
- ip6s_toomanyhdr
 - ip6stat, 110
- ip6s_tooshort
 - ip6stat, 110
- ip6s_toosmall
 - ip6stat, 110
- ip6s_total
 - ip6stat, 110
- ip6stat, 106
 - ip6_input.c, 376
 - ip6_var.h, 422
 - ip6s_badoptions, 107
 - ip6s_badscope, 107
 - ip6s_badvers, 107
 - ip6s_cantforward, 107
 - ip6s_cantfrag, 107
 - ip6s_delivered, 107
 - ip6s_exthdrtoolong, 107
 - ip6s_forward, 107
 - ip6s_forward_cachehit, 107

- ip6s_forward_cachemiss, 107
- ip6s_fragdropped, 108
- ip6s_fragmented, 108
- ip6s_fragments, 108
- ip6s_fragoverflow, 108
- ip6s_fragtimeout, 108
- ip6s_localout, 108
- ip6s_m1, 108
- ip6s_m2m, 108
- ip6s_mext1, 108
- ip6s_mext2m, 108
- ip6s_nogif, 109
- ip6s_noroute, 109
- ip6s_notmember, 109
- ip6s_nxthist, 109
- ip6s_odropped, 109
- ip6s_ofragments, 109
- ip6s_rawout, 109
- ip6s_reassembled, 109
- ip6s_redirectsent, 109
- ip6s_sources_deprecated, 109
- ip6s_sources_none, 110
- ip6s_sources_otherif, 110
- ip6s_sources_otherscope, 110
- ip6s_sources_rule, 110
- ip6s_sources_sameif, 110
- ip6s_sources_samescope, 110
- ip6s_toomanyhdr, 110
- ip6s_tooshort, 110
- ip6s_toosmall, 110
- ip6s_total, 110
- ipcomp, 112
 - comp_cpi, 112
 - comp_flags, 112
 - comp_nxt, 112
- ipcomp.h
 - __P, 425
 - IPCOMP_CPI_NEGOTIATE_MIN, 424
 - IPCOMP_DEFLATE, 424
 - IPCOMP_LZS, 424
 - IPCOMP_MAX, 425
 - IPCOMP_OUI, 425
- ipcomp6.h
 - __P, 426
- ipcomp_algorithm, 113
 - __P, 113
 - minplen, 113
- ipcomp_algorithm_lookup
 - ipcomp_core.c, 430
- ipcomp_algorithms
 - ipcomp_core.c, 430
- ipcomp_core.c
 - __P, 429
 - deflate_alloc, 429
 - deflate_common, 429
 - deflate_compress, 429
 - deflate_decompress, 430
 - deflate_free, 430
 - deflate_memlevel, 430
 - deflate_policy, 430
 - deflate_window_in, 430
 - deflate_window_out, 430
 - ipcomp_algorithm_lookup, 430
 - ipcomp_algorithms, 430
 - MOREBLOCK, 429
- IPCOMP_CPI_NEGOTIATE_MIN
 - ipcomp.h, 424
- IPCOMP_DEFLATE
 - ipcomp.h, 424
- ipcomp_input.c
 - IPLLEN_FLIPPED, 434
- IPCOMP_LZS
 - ipcomp.h, 424
- IPCOMP_MAX
 - ipcomp.h, 425
- IPCOMP_OUI
 - ipcomp.h, 425
- ipcomp_output
 - ipcomp_output.c, 437
- ipcomp_output.c
 - __P, 437
 - ipcomp_output, 437
- ipi6_addr
 - in6_pktinfo, 69
- ipi6_ifindex
 - in6_pktinfo, 69
- IPLLEN_FLIPPED
 - ah_input.c, 194
 - esp_input.c, 219
 - ipcomp_input.c, 434
- ipr_flags
 - in6_prefixreq, 72
- ipr_name
 - in6_prefixreq, 72
- ipr_origin
 - in6_prefixreq, 72
- ipr_plen
 - in6_prefixreq, 72
- ipr_pltime
 - in6_prefixreq, 72
- ipr_prefix
 - in6_prefixreq, 73
- ipr_raf_auto
 - in6_var.h, 347
- ipr_raf_onlink
 - in6_var.h, 348
- ipr_rrf_decrprefd
 - in6_var.h, 348

- ipr_rrf_decrvalid
 - in6_var.h, 348
- ipr_statef_onlink
 - in6_var.h, 348
- ipr_vltime
 - in6_prefixreq, 73
- ipsec.c
 - __P, 443
 - ip4_ah_clearartos, 453
 - ip4_ah_net_deflev, 453
 - ip4_ah_offsetmask, 454
 - ip4_ah_trans_deflev, 454
 - ip4_def_policy, 454
 - ip4_esp_net_deflev, 454
 - ip4_esp_randpad, 454
 - ip4_esp_trans_deflev, 454
 - ip4_ipsec_dfbit, 454
 - ip4_ipsec_ecn, 454
 - ipsec4_delete_pcbpolicy, 443
 - ipsec4_get_policy, 443
 - ipsec4_get_ulp, 444
 - ipsec4_getpolicybyaddr, 444
 - ipsec4_getpolicybypcb, 444
 - ipsec4_hdrsiz, 444
 - ipsec4_in_reject, 445
 - ipsec4_logpacketstr, 445
 - ipsec4_set_policy, 445
 - ipsec4_setspidx_ipaddr, 446
 - ipsec4_tunnel_validate, 446
 - ipsec_addaux, 446
 - ipsec_addhist, 446
 - ipsec_checkpcbcache, 446
 - ipsec_chkreplay, 447
 - ipsec_clearhist, 447
 - ipsec_copy_pcbpolicy, 447
 - ipsec_copypkt, 447
 - ipsec_debug, 454
 - ipsec_deepcopy_policy, 447
 - ipsec_delaux, 447
 - ipsec_delpcbpolicy, 448
 - ipsec_dumpmbuf, 448
 - ipsec_fillpcbcache, 448
 - ipsec_findaux, 448
 - ipsec_get_policy, 448
 - ipsec_get_reqllevel, 448
 - ipsec_getnhist, 449
 - ipsec_hdrsiz, 449
 - ipsec_in_reject, 449
 - ipsec_init_pcbpolicy, 449
 - ipsec_invalpcbcache, 450
 - ipsec_invalpcbcacheall, 450
 - ipsec_logsastr, 450
 - ipsec_newpcbpolicy, 450
 - ipsec_optaux, 450
 - ipsec_pcbconn, 451
 - ipsec_pcbdisconn, 451
 - ipsec_set_policy, 451
 - ipsec_setspidx, 451
 - ipsec_setspidx_mbuf, 451
 - ipsec_updatereplay, 452
 - ipsecstat, 454
 - NET_NEEDS_GIANT, 452
 - sp_cacheget, 455
 - SYSCTL_DECL, 453
 - SYSCTL_INT, 453
 - SYSCTL_STRUCT, 453
 - vshiffl, 453
- ipsec.h
 - __P, 464
 - IPSEC6CTL_NAMES, 458
 - ipsec_debug, 464
 - IPSEC_DIR_ANY, 458
 - IPSEC_DIR_INBOUND, 459
 - IPSEC_DIR_INVALID, 459
 - IPSEC_DIR_MAX, 459
 - IPSEC_DIR_OUTBOUND, 459
 - IPSEC_LEVEL_DEFAULT, 459
 - IPSEC_LEVEL_REQUIRE, 459
 - IPSEC_LEVEL_UNIQUE, 459
 - IPSEC_LEVEL_USE, 459
 - IPSEC_MANUAL_POLICYID_MAX, 459
 - IPSEC_MANUAL_REQID_MAX, 460
 - IPSEC_MODE_ANY, 460
 - IPSEC_MODE_TCPMD5, 460
 - IPSEC_MODE_TRANSPORT, 460
 - IPSEC_MODE_TUNNEL, 460
 - IPSEC_PCbsp_CONNECTED, 460
 - IPSEC_POLICY_BYPASS, 460
 - IPSEC_POLICY_DISCARD, 460
 - IPSEC_POLICY_ENTRUST, 460
 - IPSEC_POLICY_IPSEC, 461
 - IPSEC_POLICY_NONE, 461
 - IPSEC_POLICY_TCP, 461
 - IPSEC_PORT_ANY, 461
 - IPSEC_PROTO_ANY, 461
 - IPSEC_REPLAYWSIZE, 461
 - IPSEC_SPSTATE_ALIVE, 461
 - IPSEC_SPSTATE_DEAD, 461
 - IPSEC_ULPROTO_ANY, 461
 - IPSECCTL_AH_CLEARRTOS, 462
 - IPSECCTL_AH_OFFSETMASK, 462
 - IPSECCTL_DEBUG, 462
 - IPSECCTL_DEF_AH_NETLEV, 462
 - IPSECCTL_DEF_AH_TRANSLEV, 462
 - IPSECCTL_DEF_ESP_NETLEV, 462
 - IPSECCTL_DEF_ESP_TRANSLEV, 462
 - IPSECCTL_DEF_POLICY, 462
 - IPSECCTL_DFBIT, 462

- IPSECCTL_ECN, 462
- IPSECCTL_ESP_RANDPAD, 462
- IPSECCTL_MAXID, 463
- IPSECCTL_NAMES, 463
- IPSECCTL_STATS, 463
- ipseclg, 463
- ipsec4_delete_pcbpolicy
 - ipsec.c, 443
- ipsec4_get_policy
 - ipsec.c, 443
- ipsec4_get_ulp
 - ipsec.c, 444
- ipsec4_getpolicybyaddr
 - ipsec.c, 444
- ipsec4_getpolicybypcb
 - ipsec.c, 444
- ipsec4_hdrsiz
 - ipsec.c, 444
- ipsec4_in_reject
 - ipsec.c, 445
- ipsec4_logpacketstr
 - ipsec.c, 445
- ipsec4_set_policy
 - ipsec.c, 445
- ipsec4_setspidx_ipaddr
 - ipsec.c, 446
- ipsec4_tunnel_validate
 - ipsec.c, 446
- ipsec6.h
 - __P, 466
 - ip6_ah_net_deflev, 466
 - ip6_ah_trans_deflev, 466
 - ip6_def_policy, 466
 - ip6_esp_net_deflev, 466
 - ip6_esp_randpad, 466
 - ip6_esp_trans_deflev, 466
 - ip6_ipsec_ecn, 466
 - ipsec6stat, 467
- IPSEC6CTL_NAMES
 - ipsec.h, 458
- ipsec6stat
 - ipsec6.h, 467
- ipsec_addaux
 - ipsec.c, 446
- ipsec_addhist
 - ipsec.c, 446
- ipsec_checkpcbcache
 - ipsec.c, 446
- ipsec_chkreplay
 - ipsec.c, 447
- ipsec_clearhist
 - ipsec.c, 447
- ipsec_copy_pcbpolicy
 - ipsec.c, 447
- ipsec_copypkt
 - ipsec.c, 447
- ipsec_debug
 - ipsec.c, 454
 - ipsec.h, 464
- ipsec_deepcopy_policy
 - ipsec.c, 447
- ipsec_delaux
 - ipsec.c, 447
- ipsec_delpcbpolicy
 - ipsec.c, 448
- IPSEC_DIR_ANY
 - ipsec.h, 458
- IPSEC_DIR_INBOUND
 - ipsec.h, 459
- IPSEC_DIR_INVALID
 - ipsec.h, 459
- IPSEC_DIR_MAX
 - ipsec.h, 459
- IPSEC_DIR_OUTBOUND
 - ipsec.h, 459
- ipsec_dumpmbuf
 - ipsec.c, 448
- ipsec_fillpcbcache
 - ipsec.c, 448
- ipsec_findaux
 - ipsec.c, 448
- ipsec_get_policy
 - ipsec.c, 448
- ipsec_get_reqlevel
 - ipsec.c, 448
- ipsec_getnhist
 - ipsec.c, 449
- ipsec_hdrsiz
 - ipsec.c, 449
- ipsec_history, 114
- ih_proto, 114
- ih_spi, 114
- ipsec_in_reject
 - ipsec.c, 449
- ipsec_init_pcbpolicy
 - ipsec.c, 449
- ipsec_invalpcbcache
 - ipsec.c, 450
- ipsec_invalpcbcacheall
 - ipsec.c, 450
- IPSEC_LEVEL_DEFAULT
 - ipsec.h, 459
- IPSEC_LEVEL_REQUIRE
 - ipsec.h, 459
- IPSEC_LEVEL_UNIQUE
 - ipsec.h, 459
- IPSEC_LEVEL_USE
 - ipsec.h, 459

- ipsec_logsastr
 - ipsec.c, 450
- IPSEC_MANUAL_POLICYID_MAX
 - ipsec.h, 459
- IPSEC_MANUAL_REQID_MAX
 - ipsec.h, 460
- IPSEC_MODE_ANY
 - ipsec.h, 460
- IPSEC_MODE_TCPMD5
 - ipsec.h, 460
- IPSEC_MODE_TRANSPORT
 - ipsec.h, 460
- IPSEC_MODE_TUNNEL
 - ipsec.h, 460
- ipsec_newpcbpolicy
 - ipsec.c, 450
- ipsec_optaux
 - ipsec.c, 450
- ipsec_output_state, 115
 - dst, 115
 - encap, 115
 - m, 115
 - ro, 115
- ipsec_pcbconn
 - ipsec.c, 451
- ipsec_pcbdisconn
 - ipsec.c, 451
- IPSEC_PCbsp_CONNECTED
 - ipsec.h, 460
- IPSEC_POLICY_BYPASS
 - ipsec.h, 460
- IPSEC_POLICY_DISCARD
 - ipsec.h, 460
- IPSEC_POLICY_ENTRUST
 - ipsec.h, 460
- IPSEC_POLICY_IPSEC
 - ipsec.h, 461
- IPSEC_POLICY_NONE
 - ipsec.h, 461
- IPSEC_POLICY_TCP
 - ipsec.h, 461
- IPSEC_PORT_ANY
 - ipsec.h, 461
- IPSEC_PROTO_ANY
 - ipsec.h, 461
- IPSEC_REPLAYWSIZE
 - ipsec.h, 461
- ipsec_set_policy
 - ipsec.c, 451
- ipsec_setspidx
 - ipsec.c, 451
- ipsec_setspidx_mbuf
 - ipsec.c, 451
- IPSEC_SPSTATE_ALIVE
 - ipsec.h, 461
- IPSEC_SPSTATE_DEAD
 - ipsec.h, 461
- IPSEC_ULPROTO_ANY
 - ipsec.h, 461
- ipsec_updatereplay
 - ipsec.c, 452
- ipsecaux, 116
 - hdrs, 116
- IPSECCTL_AH_CLEARRTOS
 - ipsec.h, 462
- IPSECCTL_AH_OFFSETMASK
 - ipsec.h, 462
- IPSECCTL_DEBUG
 - ipsec.h, 462
- IPSECCTL_DEF_AH_NETLEV
 - ipsec.h, 462
- IPSECCTL_DEF_AH_TRANSLEV
 - ipsec.h, 462
- IPSECCTL_DEF_ESP_NETLEV
 - ipsec.h, 462
- IPSECCTL_DEF_ESP_TRANSLEV
 - ipsec.h, 462
- IPSECCTL_DEF_POLICY
 - ipsec.h, 462
- IPSECCTL_DFBIT
 - ipsec.h, 462
- IPSECCTL_ECN
 - ipsec.h, 462
- IPSECCTL_ESP_RANDPAD
 - ipsec.h, 462
- IPSECCTL_MAXID
 - ipsec.h, 463
- IPSECCTL_NAMES
 - ipsec.h, 463
- IPSECCTL_STATS
 - ipsec.h, 463
- ipseclg
 - ipsec.h, 463
- ipsecrequest, 117
 - level, 117
 - next, 117
 - saidx, 117
 - sav, 117
 - sp, 117
 - tunifp, 118
- ipsecstat, 119
 - in_ahauthfail, 119
 - in_ahauthsucc, 119
 - in_ahhist, 119
 - in_ahreplay, 120
 - in_badspi, 120
 - in_comphist, 120
 - in_espathfail, 120

- in_espauthsucc, [120](#)
- in_esphist, [120](#)
- in_espreplay, [120](#)
- in_inval, [120](#)
- in_nomem, [120](#)
- in_nosa, [120](#)
- in_polvio, [120](#)
- in_success, [121](#)
- ipsec.c, [454](#)
- out_ahhist, [121](#)
- out_comphist, [121](#)
- out_esphist, [121](#)
- out_inval, [121](#)
- out_nomem, [121](#)
- out_noroute, [121](#)
- out_nosa, [121](#)
- out_polvio, [121](#)
- out_success, [121](#)
- spdcachelookup, [122](#)
- spdcachemiss, [122](#)
- IPV6_2292DSTOPTS
 - [in6.h, 270](#)
- IPV6_2292HOPLIMIT
 - [in6.h, 271](#)
- IPV6_2292HOPOPTS
 - [in6.h, 271](#)
- IPV6_2292NEXTHOP
 - [in6.h, 271](#)
- IPV6_2292PKTINFO
 - [in6.h, 271](#)
- IPV6_2292PKTOPTIONS
 - [in6.h, 271](#)
- IPV6_2292RTHDR
 - [in6.h, 271](#)
- IPV6_ADDR_INT16_MLL
 - [in6.h, 271](#)
- IPV6_ADDR_INT16_ULL
 - [in6.h, 271](#)
- IPV6_ADDR_INT16_USL
 - [in6.h, 271](#)
- IPV6_ADDR_INT32_MLL
 - [in6.h, 272](#)
- IPV6_ADDR_INT32_MNL
 - [in6.h, 272](#)
- IPV6_ADDR_INT32_ONE
 - [in6.h, 272](#)
- IPV6_ADDR_INT32_SMP
 - [in6.h, 272](#)
- IPV6_ADDR_INT32_TWO
 - [in6.h, 272](#)
- IPV6_ADDR_MC_SCOPE
 - [in6.h, 272](#)
- IPV6_ADDR_SCOPE_GLOBAL
 - [in6.h, 272](#)
- IPV6_ADDR_SCOPE_INTFACELOCAL
 - [in6.h, 272](#)
- IPV6_ADDR_SCOPE_LINKLOCAL
 - [in6.h, 272](#)
- IPV6_ADDR_SCOPE_NODELOCAL
 - [in6.h, 273](#)
- IPV6_ADDR_SCOPE_ORGLOCAL
 - [in6.h, 273](#)
- IPV6_ADDR_SCOPE_SITELOCAL
 - [in6.h, 273](#)
- IPV6_AUTOFLOWLABEL
 - [in6.h, 273](#)
- IPV6_CHECKSUM
 - [in6.h, 273](#)
- IPV6_DEFAULT_MULTICAST_HOPS
 - [in6.h, 273](#)
- IPV6_DEFAULT_MULTICAST_LOOP
 - [in6.h, 273](#)
- IPV6_DONTFRAG
 - [in6.h, 273](#)
- IPV6_DSTOPTS
 - [in6.h, 273](#)
- IPV6_FAITH
 - [in6.h, 274](#)
- IPV6_FORWARDING
 - [ip6_var.h, 415](#)
- IPV6_FW_ADD
 - [in6.h, 274](#)
- IPV6_FW_DEL
 - [in6.h, 274](#)
- IPV6_FW_FLUSH
 - [in6.h, 274](#)
- IPV6_FW_GET
 - [in6.h, 274](#)
- IPV6_FW_ZERO
 - [in6.h, 274](#)
- IPV6_HOPLIMIT
 - [in6.h, 274](#)
- IPV6_HOPOPTS
 - [in6.h, 274](#)
- IPV6_IPSEC_POLICY
 - [in6.h, 274](#)
- IPV6_JOIN_GROUP
 - [in6.h, 275](#)
- IPV6_LEAVE_GROUP
 - [in6.h, 275](#)
- IPV6_MINMTU
 - [ip6_var.h, 415](#)
- ip6_mreq, [123](#)
 - [ip6mr_interface, 123](#)
 - [ip6mr_multiaddr, 123](#)
- IPV6_MULTICAST_HOPS
 - [in6.h, 275](#)
- IPV6_MULTICAST_IF

- in6.h, 275
- IPV6_MULTICAST_LOOP
 - in6.h, 275
- IPV6_NEXTHOP
 - in6.h, 275
- IPV6_PATHMTU
 - in6.h, 275
- IPV6_PKTINFO
 - in6.h, 275
- IPV6_PORTRANGE
 - in6.h, 276
- IPV6_PORTRANGE_DEFAULT
 - in6.h, 276
- IPV6_PORTRANGE_HIGH
 - in6.h, 276
- IPV6_PORTRANGE_LOW
 - in6.h, 276
- IPV6_PREFER_TEMPADDR
 - in6.h, 276
- IPV6_RECVDSTOPTS
 - in6.h, 276
- IPV6_RECVHOPLIMIT
 - in6.h, 276
- IPV6_RECVHOPOPTS
 - in6.h, 276
- IPV6_RECVPATHMTU
 - in6.h, 276
- IPV6_RECVPKTINFO
 - in6.h, 277
- IPV6_RECVRTHDR
 - in6.h, 277
- IPV6_RECVRTHDRDSTOPTS
 - in6.h, 277
- IPV6_RECVTCLASS
 - in6.h, 277
- IPV6_RTHDR
 - in6.h, 277
- IPV6_RTHDR_LOOSE
 - in6.h, 277
- IPV6_RTHDR_STRICT
 - in6.h, 277
- IPV6_RTHDR_TYPE_0
 - in6.h, 277
- IPV6_RTHDRDSTOPTS
 - in6.h, 277
- IPV6_SENDDIRECTS
 - in6_proto.c, 315
- IPV6_SOCKOPT_RESERVED1
 - in6.h, 278
- IPV6_TCLASS
 - in6.h, 278
- IPV6_UNICAST_HOPS
 - in6.h, 278
- IPV6_UNSPECSRC
 - ip6_var.h, 415
- IPV6_USE_MIN_MTU
 - in6.h, 278
- IPV6_V6ONLY
 - in6.h, 278
- IPV6FORWARDING
 - in6_proto.c, 315
- ipv6mr_interface
 - ipv6_mreq, 123
- ipv6mr_multiaddr
 - ipv6_mreq, 123
- irr_flags
 - in6_rrenumreq, 80
- irr_m_len
 - in6_rrenumreq, 80
- irr_m_maxlen
 - in6_rrenumreq, 80
- irr_m_minlen
 - in6_rrenumreq, 81
- irr_matchprefix
 - in6_rrenumreq, 81
- irr_name
 - in6_rrenumreq, 81
- irr_origin
 - in6_rrenumreq, 81
- irr_pltime
 - in6_rrenumreq, 81
- irr_raf_auto
 - in6_var.h, 348
- irr_raf_mask_auto
 - in6_var.h, 348
- irr_raf_mask_onlink
 - in6_var.h, 348
- irr_raf_mask_reserved
 - in6_var.h, 348
- irr_raf_onlink
 - in6_var.h, 348
- irr_raflagmask
 - in6_rrenumreq, 81
- irr_rrf
 - in6_var.h, 348
- irr_rrf_decrprefd
 - in6_var.h, 348
- irr_rrf_decrvalid
 - in6_var.h, 349
- irr_statef_onlink
 - in6_var.h, 349
- irr_u_keeplen
 - in6_rrenumreq, 81
- irr_u_uselen
 - in6_rrenumreq, 81
- irr_useprefix
 - in6_rrenumreq, 81
- irr_vltime

- in6_rrenumreq, 81
- IS2292
 - ip6_input.c, 370
- isrouter
 - in6_nbrinfo, 61
- iv
 - cblock, 25
- ivlerval
 - esp_algorithm, 29
- JUMBOOPTLEN
 - ip6_output.c, 400
- k2
 - aesxcbc_ctx, 19
- k3
 - aesxcbc_ctx, 19
- KEYDEBUG
 - ah_input.c, 194
- keymax
 - ah_algorithm, 22
 - esp_algorithm, 29
- keymin
 - ah_algorithm, 22
 - esp_algorithm, 29
- killed
 - rtqk_arg, 163
- label
 - in6_addrpolicy, 39
- last
 - nd_opts, 142
- lastused
 - secpolicy, 167
- level
 - ipsecrequest, 117
- lifetime
 - secpolicy, 167
- linkmtu
 - in6_ondireq, 65
 - nd_ifinfo, 140
- LIST_ENTRY
 - in6_multi, 57
 - in6_multi_mship, 59
 - nd_pfxrouter, 144
 - nd_prefix, 146
 - secpolicy, 167
 - secpacq, 170
- LIST_HEAD
 - in6_var.h, 356
 - ip6_moptions, 88
 - nd6.h, 506
 - nd_prefix, 146
- linfo_nd6, 124
 - ln_asked, 124
 - ln_byhint, 124
 - ln_expire, 124
 - ln_hold, 124
 - ln_next, 125
 - ln_ntick, 125
 - ln_prev, 125
 - ln_router, 125
 - ln_rt, 125
 - ln_state, 125
 - ln_timer_ch, 125
 - nd6.c, 492
 - nd6.h, 507
- llinfo_nd6, 124
 - ln_asked
 - llinfo_nd6, 124
 - ln_byhint
 - llinfo_nd6, 124
 - ln_expire
 - llinfo_nd6, 124
 - ln_hold
 - llinfo_nd6, 124
 - ln_next
 - llinfo_nd6, 125
 - ln_ntick
 - llinfo_nd6, 125
 - ln_prev
 - llinfo_nd6, 125
 - ln_router
 - llinfo_nd6, 125
 - ln_rt
 - llinfo_nd6, 125
 - ln_state
 - llinfo_nd6, 125
 - ln_timer_ch
 - llinfo_nd6, 125
- lookup_addrsel_policy
 - in6_src.c, 338
- m
 - ipsec_output_state, 115
 - rtdetq, 162
- M2MMAX
 - ip6_input.c, 370
- m6_bytes_in
 - mif6, 130
- m6_bytes_out
 - mif6, 130
- m6_flags
 - mif6, 130
- m6_ifp
 - mif6, 130
- m6_lcl_addr
 - mif6, 131
- m6_pkt_in

- mif6, 131
- m6_pkt_out
 - mif6, 131
- m6_rate_limit
 - mif6, 131
- m6_route
 - mif6, 131
- M_AUTHIPDGM
 - in6.h, 278
- M_AUTHIPHDR
 - in6.h, 278
- M_DECRYPTED
 - in6.h, 278
- M_HASCL
 - ip6_mroute.c, 381
- M_LOOP
 - in6.h, 278
- MAKE_CHAIN
 - ip6_output.c, 400
- MAKE_EXTHDR
 - ip6_output.c, 400
- MALLOC_DEFINE
 - frag6.c, 232
 - in6.c, 260
 - ip6_mroute.c, 387
 - ip6_output.c, 409
- match_addrsel_policy
 - in6_src.c, 338
- MAX_RANDOM_FACTOR
 - nd6.h, 498
- MAX_REACHABLE_TIME
 - nd6.h, 499
- MAX_RTR_SOLICITATION_DELAY
 - nd6.h, 499
- MAX_RTR_SOLICITATIONS
 - nd6.h, 499
- MAX_TEMP_DESYNC_FACTOR
 - nd6.h, 499
- MAX_UPQ6
 - ip6_mroute.h, 394
- MAXIVLEN
 - esp_core.c, 211
- MAXMIFS
 - ip6_mroute.h, 394
- maxmtu
 - in6_ondireq, 65
 - nd_ifinfo, 141
- MC6_SEND
 - ip6_mroute.c, 381
- MD5_RESULTLEN
 - ah_core.c, 187
- mf6c, 126
 - mf6c_byte_cnt, 126
 - mf6c_expire, 126
 - mf6c_ifset, 126
 - mf6c_last_assert, 127
 - mf6c_mcastgrp, 127
 - mf6c_next, 127
 - mf6c_origin, 127
 - mf6c_parent, 127
 - mf6c_pkt_cnt, 127
 - mf6c_stall, 127
 - mf6c_wrong_if, 127
- mf6c_byte_cnt
 - mf6c, 126
- mf6c_expire
 - mf6c, 126
- mf6c_ifset
 - mf6c, 126
- MF6C_INCOMPLETE_PARENT
 - ip6_mroute.h, 394
- mf6c_last_assert
 - mf6c, 127
- mf6c_mcastgrp
 - mf6c, 127
- mf6c_next
 - mf6c, 127
- mf6c_origin
 - mf6c, 127
- mf6c_parent
 - mf6c, 127
- mf6c_pkt_cnt
 - mf6c, 127
- mf6c_stall
 - mf6c, 127
- mf6c_wrong_if
 - mf6c, 127
- mf6cc_ifset
 - mf6cctl, 129
- mf6cc_mcastgrp
 - mf6cctl, 129
- mf6cc_origin
 - mf6cctl, 129
- mf6cc_parent
 - mf6cctl, 129
- mf6cctl, 129
 - mf6cc_ifset, 129
 - mf6cc_mcastgrp, 129
 - mf6cc_origin, 129
 - mf6cc_parent, 129
- MF6CFIND
 - ip6_mroute.c, 381
- MF6CHASH
 - ip6_mroute.c, 382
- MF6CHASHMOD
 - ip6_mroute.h, 394
- mf6ctable
 - ip6_mroute.c, 390

- MF6CTBSIZ
 - ip6_mroute.h, 394
- mif6, 130
 - m6_bytes_in, 130
 - m6_bytes_out, 130
 - m6_flags, 130
 - m6_ifp, 130
 - m6_lcl_addr, 131
 - m6_pkt_in, 131
 - m6_pkt_out, 131
 - m6_rate_limit, 131
 - m6_route, 131
- mif6c_flags
 - mif6ctl, 132
- mif6c_mifi
 - mif6ctl, 132
- mif6c_pifi
 - mif6ctl, 132
- mif6ctl, 132
 - mif6c_flags, 132
 - mif6c_mifi, 132
 - mif6c_pifi, 132
- mif6table
 - ip6_mroute.c, 390
- MIFF_REGISTER
 - ip6_mroute.h, 394
- mifi
 - sioc_mif_req6, 171
- mifi_t
 - ip6_mroute.h, 396
- MIN_RANDOM_FACTOR
 - nd6.h, 499
- minplen
 - ipcomp_algorithm, 113
- mld6.c
 - in6_addmulti, 470
 - in6_delmulti, 470
 - ip6_opts, 474
 - mld6_init, 471
 - mld6_input, 471
 - mld6_sendpkt, 471
 - mld6_start_listening, 472
 - mld6_stop_listening, 473
 - mld_starttimer, 473
 - mld_stoptimer, 473
 - mld_timeo, 474
 - MLD_TIMER_SCALE, 470
 - mld_timerresid, 474
 - MLD_UNSOLICITED_REPORT_INTERVAL, 470
- mld6_fasttimeo
 - mld6_var.h, 476
- mld6_init
 - mld6.c, 471
 - mld6_var.h, 476
- mld6_input
 - mld6.c, 471
 - mld6_var.h, 476
- mld6_sendpkt
 - mld6.c, 471
- mld6_start_listening
 - mld6.c, 472
 - mld6_var.h, 476
- mld6_stop_listening
 - mld6.c, 473
 - mld6_var.h, 477
- mld6_var.h
 - mld6_fasttimeo, 476
 - mld6_init, 476
 - mld6_input, 476
 - mld6_start_listening, 476
 - mld6_stop_listening, 477
 - MLD_IREPORTEDLAST, 475
 - MLD_OTHERLISTENER, 475
 - MLD_RANDOM_DELAY, 475
 - MLD_REPORTPENDING, 475
 - MLD_IREPORTEDLAST
 - mld6_var.h, 475
 - MLD_OTHERLISTENER
 - mld6_var.h, 475
 - MLD_RANDOM_DELAY
 - mld6_var.h, 475
 - MLD_REPORTPENDING
 - mld6_var.h, 475
- mld_starttimer
 - mld6.c, 473
- mld_stoptimer
 - mld6.c, 473
- mld_timeo
 - mld6.c, 474
- MLD_TIMER_SCALE
 - mld6.c, 470
- mld_timerresid
 - mld6.c, 474
- MLD_UNSOLICITED_REPORT_INTERVAL
 - mld6.c, 470
- MLTMASK_LEN
 - in6.c, 249
- mode
 - rtqk_arg, 163
- MOREBLOCK
 - ipcomp_core.c, 429
- MRT6_ADD_MFC
 - ip6_mroute.h, 394
- MRT6_ADD_MIF
 - ip6_mroute.h, 394
- MRT6_DEL_MFC
 - ip6_mroute.h, 394

- MRT6_DEL_MIF
 - ip6_mroute.h, 395
- MRT6_DONE
 - ip6_mroute.h, 395
- MRT6_INIT
 - ip6_mroute.h, 395
- mrt6_ioctl
 - ip6_mroute.c, 387
- MRT6_OINIT
 - ip6_mroute.h, 395
- MRT6_PIM
 - ip6_mroute.h, 395
- mrt6msg, 133
 - im6_dst, 133
 - im6_mbz, 133
 - im6_mif, 133
 - im6_msgtype, 133
 - im6_pad, 133
- MRT6MSG_NOCACHE
 - ip6_mroute.h, 395
- MRT6MSG_WHOLEPKT
 - ip6_mroute.h, 395
- MRT6MSG_WRONGMIF
 - ip6_mroute.h, 395
- mrt6s_bad_tunnel
 - mrt6stat, 135
- mrt6s_cache_cleanups
 - mrt6stat, 135
- mrt6s_cant_tunnel
 - mrt6stat, 135
- mrt6s_drop_sel
 - mrt6stat, 135
- mrt6s_mfc_lookups
 - mrt6stat, 135
- mrt6s_mfc_misses
 - mrt6stat, 135
- mrt6s_no_route
 - mrt6stat, 136
- mrt6s_pkt2large
 - mrt6stat, 136
- mrt6s_q_overflow
 - mrt6stat, 136
- mrt6s_upcalls
 - mrt6stat, 136
- mrt6s_upq_ovflw
 - mrt6stat, 136
- mrt6s_upq_sockfull
 - mrt6stat, 136
- mrt6s_wrong_if
 - mrt6stat, 136
- mrt6stat, 135
 - ip6_mroute.c, 390
 - mrt6s_bad_tunnel, 135
 - mrt6s_cache_cleanups, 135
 - mrt6s_cant_tunnel, 135
 - mrt6s_drop_sel, 135
 - mrt6s_mfc_lookups, 135
 - mrt6s_mfc_misses, 135
 - mrt6s_no_route, 136
 - mrt6s_pkt2large, 136
 - mrt6s_q_overflow, 136
 - mrt6s_upcalls, 136
 - mrt6s_upq_ovflw, 136
 - mrt6s_upq_sockfull, 136
 - mrt6s_wrong_if, 136
- mtu
 - nd_opts, 142
- mtuex_arg, 137
 - nextstop, 137
 - rnh, 137
- MTUTIMO_DEFAULT
 - in6_rmx.c, 326
- multicast_register_if6
 - ip6_mroute.c, 390
- n6expire
 - ip6_mroute.c, 390
- name
 - ah_algorithm, 22
 - esp_algorithm, 29
- ND
 - nd6.c, 481
- nd6.c
 - __P, 482
 - all1_sa, 492
 - clear_llinfo_pqueue, 482
 - in6_tmpaddrtimer_ch, 492
 - llinfo_nd6, 492
 - ND, 481
 - nd6_allocated, 492
 - nd6_cache_lladdr, 482
 - nd6_debug, 492
 - nd6_delay, 492
 - nd6_free, 483
 - nd6_gctimer, 492
 - nd6_ifattach, 483
 - nd6_ifdetach, 484
 - nd6_init, 484
 - nd6_inuse, 493
 - nd6_ioctl, 484
 - nd6_is_addr_neighbor, 485
 - nd6_is_new_addr_neighbor, 485
 - nd6_llinfo_settimer, 486
 - nd6_llinfo_timer, 486
 - nd6_lookup, 487
 - nd6_maxndopt, 493
 - nd6_maxnudhint, 493
 - nd6_maxqueuelen, 493

- nd6_mmaxtries, 493
- nd6_need_cache, 487
- nd6_nud_hint, 488
- nd6_option, 488
- nd6_option_init, 488
- nd6_options, 488
- nd6_output, 488
- nd6_prune, 493
- nd6_purge, 489
- ND6_RECALC_REACHTM_INTERVAL, 481
- nd6_recalc_reachtm_interval, 493
- nd6_rtrequest, 489
- nd6_setmtu, 489
- nd6_setmtu0, 490
- ND6_SLOWTIMER_INTERVAL, 481
- nd6_slowtimo, 490
- nd6_slowtimo_ch, 493
- nd6_storelladdr, 490
- nd6_sysctl_drlist, 490
- nd6_sysctl_prlist, 490
- nd6_timer, 491
- nd6_timer_ch, 493
- nd6_umaxtries, 494
- nd6_uselookback, 494
- nd_defrouter, 494
- nd_prefix, 494
- regen_tmpaddr, 491
- RTRADDR, 481
- SDL, 481
- senderr, 482
- SIN6, 482
- SYSCTL_INT, 492
- SYSCTL_NODE, 492
- nd6.h
 - __P, 506
 - DEF_TEMP_PREFERRED_LIFETIME, 498
 - DEF_TEMP_VALID_LIFETIME, 498
 - DRLSTSIZ, 498
 - IN6_LINKMTU, 498
 - ip6_desync_factor, 506
 - ip6_temp_preferred_lifetime, 507
 - ip6_temp_regen_advance, 507
 - ip6_temp_valid_lifetime, 507
 - LIST_HEAD, 506
 - llinfo_nd6, 507
 - MAX_RANDOM_FACTOR, 498
 - MAX_REACHABLE_TIME, 499
 - MAX_RTR_SOLICITATION_DELAY, 499
 - MAX_RTR_SOLICITATIONS, 499
 - MAX_TEMP_DESYNC_FACTOR, 499
 - MIN_RANDOM_FACTOR, 499
 - nd6_debug, 507
 - nd6_defifindex, 507
 - nd6_delay, 507
 - nd6_gctimer, 507
 - ND6_IFF_ACCEPT_RTADV, 499
 - ND6_IFF_DONT_SET_IFROUTE, 499
 - ND6_IFF_IFDISABLED, 499
 - ND6_IFF_PERFORMNUD, 499
 - ND6_IFF_PREFER_SOURCE, 499
 - ND6_INFINITE_LIFETIME, 500
 - ND6_IS_LLINFO_PROBREACH, 500
 - ND6_LLINFO_DELAY, 500
 - ND6_LLINFO_INCOMPLETE, 500
 - ND6_LLINFO_NOSTATE, 500
 - ND6_LLINFO_PERMANENT, 500
 - ND6_LLINFO_PROBE, 500
 - ND6_LLINFO_REACHABLE, 500
 - ND6_LLINFO_STALE, 501
 - nd6_maxnudhint, 507
 - nd6_mmaxtries, 508
 - nd6_prune, 508
 - nd6_timer_ch, 508
 - nd6_umaxtries, 508
 - nd6_uselookback, 508
 - nd6log, 501
 - ND_COMPUTE_RUNTIME, 501
 - nd_defrouter, 508
 - ND_IFINFO, 501
 - nd_opts_done, 501
 - nd_opts_last, 501
 - nd_opts_mtu, 501
 - nd_opts_pi, 502
 - nd_opts_pi_end, 502
 - nd_opts_rh, 502
 - nd_opts_search, 502
 - nd_opts_src_lladdr, 502
 - nd_opts_tgt_lladdr, 502
 - nd_prefix, 508
 - ndpr_next, 502
 - ndpr_raf, 502
 - ndpr_raf_auto, 502
 - ndpr_raf_onlink, 502
 - ndpr_raf_router, 502
 - NDPRF_DETACHED, 503
 - NDPRF_ONLINK, 503
 - pfr_next, 503
 - PRLSTSIZ, 503
 - prm_raf_auto, 503
 - prm_raf_onlink, 503
 - prm_rrf_decrprefd, 503
 - prm_rrf_decrvalid, 503
 - prm_statef_onlink, 503
 - REACHABLE_TIME, 503
 - RETRANS_TIMER, 504
 - RTF_ANNOUNCE, 504
 - RTR_SOLICITATION_INTERVAL, 504

- TAILQ_HEAD, 506
- TEMPADDR_REGEN_ADVANCE, 504
- nd6_allocated
 - nd6.c, 492
- nd6_cache_lladdr
 - nd6.c, 482
- nd6_dad_duplicated
 - nd6_nbr.c, 512
- nd6_dad_find
 - nd6_nbr.c, 512
- nd6_dad_na_input
 - nd6_nbr.c, 512
- nd6_dad_ns_input
 - nd6_nbr.c, 513
- nd6_dad_ns_output
 - nd6_nbr.c, 513
- nd6_dad_start
 - nd6_nbr.c, 513
- nd6_dad_starttimer
 - nd6_nbr.c, 514
- nd6_dad_stop
 - nd6_nbr.c, 514
- nd6_dad_stoptimer
 - nd6_nbr.c, 515
- nd6_dad_timer
 - nd6_nbr.c, 515
- nd6_debug
 - nd6.c, 492
 - nd6.h, 507
- nd6_defindex
 - nd6.h, 507
 - nd6_rtr.c, 534
- nd6_defifp
 - nd6_rtr.c, 534
- nd6_delay
 - nd6.c, 492
 - nd6.h, 507
- nd6_free
 - nd6.c, 483
- nd6_gctimer
 - nd6.c, 492
 - nd6.h, 507
- nd6_ifattach
 - nd6.c, 483
- nd6_ifdetach
 - nd6.c, 484
- ND6_IFF_ACCEPT_RTADV
 - nd6.h, 499
- ND6_IFF_DONT_SET_IFROUTE
 - nd6.h, 499
- ND6_IFF_IFDISABLED
 - nd6.h, 499
- ND6_IFF_PERFORMNUD
 - nd6.h, 499
- ND6_IFF_PREFER_SOURCE
 - nd6.h, 499
- nd6_ifptomac
 - nd6_nbr.c, 515
- ND6_INFINITE_LIFETIME
 - nd6.h, 500
- nd6_init
 - nd6.c, 484
- nd6_inuse
 - nd6.c, 493
- nd6_ioctl
 - nd6.c, 484
- nd6_is_addr_neighbor
 - nd6.c, 485
- ND6_IS_LLINFO_PROBREACH
 - nd6.h, 500
- nd6_is_new_addr_neighbor
 - nd6.c, 485
- ND6_LLINFO_DELAY
 - nd6.h, 500
- ND6_LLINFO_INCOMPLETE
 - nd6.h, 500
- ND6_LLINFO_NOSTATE
 - nd6.h, 500
- ND6_LLINFO_PERMANENT
 - nd6.h, 500
- ND6_LLINFO_PROBE
 - nd6.h, 500
- ND6_LLINFO_REACHABLE
 - nd6.h, 500
- nd6_llinfo_settimer
 - nd6.c, 486
- ND6_LLINFO_STALE
 - nd6.h, 501
- nd6_llinfo_timer
 - nd6.c, 486
- nd6_lookup
 - nd6.c, 487
- nd6_maxndopt
 - nd6.c, 493
- nd6_maxnudhint
 - nd6.c, 493
 - nd6.h, 507
- nd6_maxqueuelen
 - nd6.c, 493
- nd6_mmaxtries
 - nd6.c, 493
 - nd6.h, 508
- nd6_na_input
 - nd6_nbr.c, 515
- nd6_na_output
 - nd6_nbr.c, 516
- nd6_nbr.c
 - __P, 512

- dad_ignore_ns, 518
- dad_init, 518
- dad_maxtry, 518
- dadq, 518
- nd6_dad_duplicated, 512
- nd6_dad_find, 512
- nd6_dad_na_input, 512
- nd6_dad_ns_input, 513
- nd6_dad_ns_output, 513
- nd6_dad_start, 513
- nd6_dad_starttimer, 514
- nd6_dad_stop, 514
- nd6_dad_stoptimer, 515
- nd6_dad_timer, 515
- nd6_ifptomac, 515
- nd6_na_input, 515
- nd6_na_output, 516
- nd6_ns_input, 517
- nd6_ns_output, 517
- SDL, 511
- TAILQ_HEAD, 518
- nd6_need_cache
 - nd6.c, 487
- nd6_ns_input
 - nd6_nbr.c, 517
- nd6_ns_output
 - nd6_nbr.c, 517
- nd6_nud_hint
 - nd6.c, 488
- nd6_option
 - nd6.c, 488
- nd6_option_init
 - nd6.c, 488
- nd6_options
 - nd6.c, 488
- nd6_output
 - nd6.c, 488
- nd6_prefix_lookup
 - nd6_rtr.c, 527
- nd6_prefix_offlink
 - nd6_rtr.c, 527
- nd6_prefix_onlink
 - nd6_rtr.c, 528
- nd6_prelist_add
 - nd6_rtr.c, 528
- nd6_prune
 - nd6.c, 493
 - nd6.h, 508
- nd6_purge
 - nd6.c, 489
- nd6_ra_input
 - nd6_rtr.c, 529
- ND6_RECALC_REACHTM_INTERVAL
 - nd6.c, 481
- nd6_recalc_reachtm_interval
 - nd6.c, 493
 - nd6_rtr.c, 534
- nd6_rs_input
 - nd6_rtr.c, 529
- nd6_rtmsg
 - nd6_rtr.c, 530
- nd6_rtr.c
 - __P, 523
 - defrouter_addrq, 523
 - defrouter_delreq, 523
 - defrouter_lookup, 523
 - defrouter_reset, 524
 - defrouter_select, 524
 - defrtrlist_del, 524
 - defrtrlist_update, 525
 - find_pfxlist_reachable_router, 525
 - in6_ifadd, 526
 - in6_init_address_ltimes, 526
 - in6_init_prefix_ltimes, 527
 - in6_tmpifadd, 527
 - ip6_desync_factor, 534
 - ip6_temp_preferred_lifetime, 534
 - ip6_temp_regen_advance, 534
 - ip6_temp_valid_lifetime, 534
 - ip6_use_tempaddr, 534
 - nd6_defifindex, 534
 - nd6_defifp, 534
 - nd6_prefix_lookup, 527
 - nd6_prefix_offlink, 527
 - nd6_prefix_onlink, 528
 - nd6_prelist_add, 528
 - nd6_ra_input, 529
 - nd6_recalc_reachtm_interval, 534
 - nd6_rs_input, 529
 - nd6_rtmsg, 530
 - nd6_setdefaultiface, 530
 - pfxlist_onlink_check, 530
 - pfxrtr_add, 531
 - pfxrtr_del, 531
 - pfxrtr_lookup, 531
 - prelist_remove, 531
 - prelist_update, 532
 - rt6_deleteroute, 533
 - rt6_flush, 533
 - rtpref, 533
 - RTPREF_HIGH, 522
 - RTPREF_INVALID, 522
 - RTPREF_LOW, 522
 - RTPREF_MEDIUM, 522
 - RTPREF_RESERVED, 522
 - SDL, 522
 - SIN6, 522
 - TWOHOUR, 522

- nd6_rtrequest
 - nd6.c, 489
- nd6_setdefaultiface
 - nd6_rtr.c, 530
- nd6_setmtu
 - nd6.c, 489
- nd6_setmtu0
 - nd6.c, 490
- ND6_SLOWTIMER_INTERVAL
 - nd6.c, 481
- nd6_slowtimo
 - nd6.c, 490
- nd6_slowtimo_ch
 - nd6.c, 493
- nd6_storelladdr
 - nd6.c, 490
- nd6_sysctl_drlist
 - nd6.c, 490
- nd6_sysctl_prlist
 - nd6.c, 490
- nd6_timer
 - nd6.c, 491
- nd6_timer_ch
 - nd6.c, 493
 - nd6.h, 508
- nd6_umaxtries
 - nd6.c, 494
 - nd6.h, 508
- nd6_uselookback
 - nd6.c, 494
 - nd6.h, 508
- nd6log
 - nd6.h, 501
- ND_COMPUTE_RUNTIME
 - nd6.h, 501
- nd_defrouter, 138
 - expire, 138
 - flags, 138
 - ifp, 138
 - installed, 139
 - nd6.c, 494
 - nd6.h, 508
 - rtaddr, 139
 - rtlifetime, 139
 - TAILQ_ENTRY, 138
- ND_IFINFO
 - nd6.h, 501
- nd_ifinfo, 140
 - basereachable, 140
 - chlim, 140
 - flags, 140
 - in6_ifextra, 50
 - initialized, 140
 - linkmtu, 140
 - maxmtu, 141
 - randomid, 141
 - randomseed0, 141
 - randomseed1, 141
 - reachable, 141
 - recalctm, 141
 - retrans, 141
- nd_opt_array
 - nd_opts, 142
- nd_opt_each
 - nd_opts, 142
- nd_opts, 142
 - done, 142
 - last, 142
 - mtu, 142
 - nd_opt_array, 142
 - nd_opt_each, 142
 - pi_beg, 142
 - pi_end, 142
 - rh, 143
 - search, 143
 - src_lladdr, 143
 - tgt_lladdr, 143
 - zero, 143
- nd_opts_done
 - nd6.h, 501
- nd_opts_last
 - nd6.h, 501
- nd_opts_mtu
 - nd6.h, 501
- nd_opts_pi
 - nd6.h, 502
- nd_opts_pi_end
 - nd6.h, 502
- nd_opts_rh
 - nd6.h, 502
- nd_opts_search
 - nd6.h, 502
- nd_opts_src_lladdr
 - nd6.h, 502
- nd_opts_tgt_lladdr
 - nd6.h, 502
- nd_pfxrouter, 144
 - LIST_ENTRY, 144
 - router, 144
- nd_prefix, 145
 - LIST_ENTRY, 146
 - LIST_HEAD, 146
 - nd6.c, 494
 - nd6.h, 508
 - ndpr_expire, 146
 - ndpr_flags, 146
 - ndpr_ifp, 146
 - ndpr_lastupdate, 146

- ndpr_mask, 146
- ndpr_plen, 146
- ndpr_pltime, 146
- ndpr_preferred, 146
- ndpr_prefix, 147
- ndpr_refcnt, 147
- ndpr_stateflags, 147
- ndpr_vltime, 147
- nd_prefixctl, 148
 - ndpr_flags, 148
 - ndpr_ifp, 148
 - ndpr_plen, 148
 - ndpr_pltime, 148
 - ndpr_prefix, 148
 - ndpr_vltime, 149
- ndi
 - in6_ndireq, 64
 - in6_ondireq, 66
- ndpr_expire
 - nd_prefix, 146
- ndpr_flags
 - nd_prefix, 146
 - nd_prefixctl, 148
- ndpr_ifp
 - nd_prefix, 146
 - nd_prefixctl, 148
- ndpr_lastupdate
 - nd_prefix, 146
- ndpr_mask
 - nd_prefix, 146
- ndpr_next
 - nd6.h, 502
- ndpr_plen
 - nd_prefix, 146
 - nd_prefixctl, 148
- ndpr_pltime
 - nd_prefix, 146
 - nd_prefixctl, 148
- ndpr_preferred
 - nd_prefix, 146
- ndpr_prefix
 - nd_prefix, 147
 - nd_prefixctl, 148
- ndpr_raf
 - nd6.h, 502
- ndpr_raf_auto
 - nd6.h, 502
- ndpr_raf_onlink
 - nd6.h, 502
- ndpr_raf_router
 - nd6.h, 502
- ndpr_refcnt
 - nd_prefix, 147
- ndpr_stateflags
 - nd_prefix, 147
- ndpr_vltime
 - nd_prefix, 147
- NDPRF_DETACHED
 - nd6.h, 503
- NDPRF_ONLINK
 - nd6.h, 503
- NET_NEEDS_GIANT
 - ipsec.c, 452
- newah, 150
 - ah_len, 150
 - ah_nxt, 150
 - ah_reserve, 150
 - ah_seq, 150
 - ah_spi, 150
- newesp, 151
 - esp_seq, 151
 - esp_spi, 151
- NEXT
 - in6_src.c, 334
- next
 - ipsecrequest, 117
 - rtdetq, 162
- nextstop
 - mtuex_arg, 137
 - rtqk_arg, 163
- ni6_addr
 - icmp6.c, 241
- ni6_dnsmatch
 - icmp6.c, 242
- ni6_input
 - icmp6.c, 242
- ni6_nametodns
 - icmp6.c, 242
- ni6_store_addr
 - icmp6.c, 242
- NIFBITS
 - ip6_mroute.h, 395
- NO_RTE_FOUND
 - ip6_mroute.c, 382
- nonce
 - cblock, 25
- NONCESIZE
 - esp_aesctr.c, 205
- notreviewed.dox, 177
- nousrreqs
 - in6_proto.c, 322
- nummifs
 - ip6_mroute.c, 390
- obytes
 - sioc_mif_req6, 171
- ocount

- sioc_mif_req6, 171
- omrt6msg, 152
 - im6_dst, 152
 - im6_mbz, 152
 - im6_mif, 152
 - im6_msgtype, 152
 - unused1, 152
 - unused2, 153
- onlink
 - in6_prflags::prf_ra, 76
 - in6_rrenumreq::irr_raflagmask, 82
- OPTBIT
 - ip6_output.c, 400
- OPTSET
 - ip6_output.c, 400
- OPTSET2292
 - ip6_output.c, 401
- origin
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- OSIOCGIFINFO_IN6
 - in6_var.h, 349
- out_ahhist
 - ipsestat, 121
- out_comphist
 - ipsestat, 121
- out_esphist
 - ipsestat, 121
- out_inval
 - ipsestat, 121
- out_nomem
 - ipsestat, 121
- out_noroute
 - ipsestat, 121
- out_nosa
 - ipsestat, 121
- out_polvio
 - ipsestat, 121
- out_success
 - ipsestat, 121
- padbound
 - esp_algorithm, 29
- persist
 - secpolicy, 167
- pfacts
 - randomtab, 157
- pfr_next
 - nd6.h, 503
- pfxlist_onlink_check
 - nd6_rtr.c, 530
- pfxrtr_add
 - nd6_rtr.c, 531
- pfxrtr_del
 - nd6_rtr.c, 531
- pfxrtr_lookup
 - nd6_rtr.c, 531
- phyint_send
 - ip6_mroute.c, 388
- pi_beg
 - nd_opts, 142
- pi_end
 - nd_opts, 142
- pim, 154
 - pim_cksum, 154
 - pim_rsv, 154
 - pim_type, 154
 - pim_ver, 154
- pim6
 - ip6_mroute.c, 390
- pim6.h
 - PIM6_REG_MINLEN, 535
 - PIM_MINLEN, 535
 - PIM_NULL_REGISTER, 535
 - PIM_REGISTER, 535
 - PIM_VERSION, 535
- PIM6_CHECKSUM
 - ip6_mroute.c, 382
- pim6_input
 - ip6_mroute.c, 388
- PIM6_REG_MINLEN
 - pim6.h, 535
- pim6_var.h
 - __P, 536
 - PIM6CTL_MAXID, 536
 - PIM6CTL_NAMES, 536
 - PIM6CTL_STATS, 536
- PIM6CTL_MAXID
 - pim6_var.h, 536
- PIM6CTL_NAMES
 - pim6_var.h, 536
- PIM6CTL_STATS
 - pim6_var.h, 536
- pim6s_rcv_badregisters
 - pim6stat, 155
- pim6s_rcv_badsum
 - pim6stat, 155
- pim6s_rcv_badversion
 - pim6stat, 155
- pim6s_rcv_registers
 - pim6stat, 155
- pim6s_rcv_tooshort
 - pim6stat, 155
- pim6s_rcv_total
 - pim6stat, 155
- pim6s_snd_registers
 - pim6stat, 156

- pim6stat, 155
 - ip6_mroute.c, 391
 - pim6s_rcv_badregisters, 155
 - pim6s_rcv_badsum, 155
 - pim6s_rcv_badversion, 155
 - pim6s_rcv_registers, 155
 - pim6s_rcv_tooshort, 155
 - pim6s_rcv_total, 155
 - pim6s_snd_registers, 156
- pim_cksum
 - pim, 154
- PIM_MINLEN
 - pim6.h, 535
- PIM_NULL_REGISTER
 - pim6.h, 535
- PIM_REGISTER
 - pim6.h, 535
- pim_rsv
 - pim, 154
- pim_type
 - pim, 154
- pim_ver
 - pim, 154
- PIM_VERSION
 - pim6.h, 535
- pktcnt
 - sioc_sg_req6, 172
- PKTOPT_EXTHDRCPY
 - ip6_output.c, 401
- pltime
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- pmod
 - ip6_id.c, 365
- pmtu_expire
 - in6_proto.c, 322
- pmtu_probe
 - in6_proto.c, 323
- policy
 - secpolicy, 167
- PR_ABRTACPTDIS
 - in6_proto.c, 315
- pr_domain
 - ip6protosw, 102
- pr_flags
 - ip6protosw, 102
- PR_LISTEN
 - in6_proto.c, 315
- PR_ORIG_KERNEL
 - in6_var.h, 349
- PR_ORIG_RA
 - in6_var.h, 349
- PR_ORIG_RR
 - in6_var.h, 349
- PR_ORIG_STATIC
 - in6_var.h, 349
- pr_protocol
 - ip6protosw, 102
- pr_type
 - ip6protosw, 102
- pr_usrreqs
 - ip6protosw, 102
- preced
 - in6_addrpolicy, 39
- prefd
 - secpolicyindex, 169
- prefix
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- prefixlen
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- prefs
 - secpolicyindex, 169
- prelist_remove
 - nd6_rtr.c, 531
- prelist_update
 - nd6_rtr.c, 532
- prf_ra
 - in6_prflags, 74
- prf_reserved1
 - in6_prflags, 74
- prf_reserved2
 - in6_prflags, 74
- prf_reserved3
 - in6_prflags, 74
- prf_reserved4
 - in6_prflags, 74
- prf_rr
 - in6_prflags, 75
- priv
 - inpcbpolicy, 86
- PRLSTSIZ
 - nd6.h, 503
- prm_expire
 - inet6_ndpr_msghdr, 84
- prm_flags
 - inet6_ndpr_msghdr, 84
- prm_index
 - inet6_ndpr_msghdr, 84
- prm_plen
 - inet6_ndpr_msghdr, 84
- prm_pltime
 - inet6_ndpr_msghdr, 84
- prm_preferred

- inet6_ndpr_msghdr, 84
- prm_raf_auto
 - nd6.h, 503
- prm_raf_onlink
 - nd6.h, 503
- prm_rrf_decrprefd
 - nd6.h, 503
- prm_rrf_decrvalid
 - nd6.h, 503
- prm_statef_onlink
 - nd6.h, 503
- prm_vltim
 - inet6_ndpr_msghdr, 84
- r_ek
 - aesctr_ctx, 18
- r_k1s
 - aesxcbc_ctx, 19
- r_k2s
 - aesxcbc_ctx, 20
- r_k3s
 - aesxcbc_ctx, 20
- r_nr
 - aesctr_ctx, 18
 - aesxcbc_ctx, 20
- raflags
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- randomid
 - ip6_id.c, 365
 - nd_ifinfo, 141
- randomseed0
 - nd_ifinfo, 141
- randomseed1
 - nd_ifinfo, 141
- randomtab, 157
 - pfacts, 157
 - ru_a, 157
 - ru_agen, 157
 - ru_b, 157
 - ru_bits, 157
 - ru_counter, 158
 - ru_g, 158
 - ru_gen, 158
 - ru_m, 158
 - ru_max, 158
 - ru_msb, 158
 - ru_n, 158
 - ru_out, 158
 - ru_reseed, 158
 - ru_seed, 159
 - ru_seed2, 159
 - ru_x, 159
- randomtab_20
 - ip6_id.c, 365
- randomtab_32
 - ip6_id.c, 365
- raw_ip6.c
 - ifatoia6, 539
 - rip6_abort, 539
 - rip6_attach, 539
 - rip6_bind, 540
 - rip6_close, 540
 - rip6_connect, 540
 - rip6_ctlinput, 540
 - rip6_ctloutput, 541
 - rip6_detach, 541
 - rip6_disconnect, 542
 - rip6_input, 542
 - rip6_output, 542
 - rip6_send, 543
 - rip6_shutdown, 543
 - rip6_usrreqs, 544
 - rip6stat, 544
 - rip_recvspace, 544
 - rip_sendspace, 544
 - ripcb, 544
 - ripcbinfo, 544
 - satosin6, 539
- raw_ip6.h
 - rip6stat, 545
- reachable
 - in6_ondireq, 66
 - nd_ifinfo, 141
- REACHABLE_TIME
 - nd6.h, 503
- readonly
 - secpolicy, 167
- recalctm
 - in6_ondireq, 66
 - nd_ifinfo, 141
- receivedra
 - in6_ondireq, 66
- REDUCE
 - in6_cksum.c, 282
- refcnt
 - in6_prefix, 71
 - secpolicy, 167
- reg_mif_num
 - ip6_mroute.c, 391
- regen_tmpaddr
 - nd6.c, 491
- register_send
 - ip6_mroute.c, 389
- REPLACE
 - in6_src.c, 334
- req

- secpolicy, 168
- reserved
 - in6_prflags::prf_ra, 76
 - in6_prflags::prf_rr, 77
 - in6_rrenumreq::irr_raflagmask, 82
- retrans
 - in6_ondireq, 66
 - nd_ifinfo, 141
- RETRANS_TIMER
 - nd6.h, 504
- rh
 - nd_opts, 143
- rip6_abort
 - raw_ip6.c, 539
- rip6_attach
 - raw_ip6.c, 539
- rip6_bind
 - raw_ip6.c, 540
- rip6_close
 - raw_ip6.c, 540
- rip6_connect
 - raw_ip6.c, 540
- rip6_ctlinput
 - raw_ip6.c, 540
- rip6_ctloutput
 - raw_ip6.c, 541
- rip6_detach
 - raw_ip6.c, 541
- rip6_disconnect
 - raw_ip6.c, 542
- rip6_input
 - raw_ip6.c, 542
- rip6_output
 - raw_ip6.c, 542
- rip6_recvspace
 - in6_proto.c, 323
- rip6_send
 - raw_ip6.c, 543
- rip6_sendspace
 - in6_proto.c, 323
- rip6_shutdown
 - raw_ip6.c, 543
- rip6_usrreqs
 - ip6_var.h, 422
 - raw_ip6.c, 544
- rip6s_badsum
 - rip6stat, 160
- rip6s_fullsock
 - rip6stat, 160
- rip6s_ipackets
 - rip6stat, 160
- rip6s_isum
 - rip6stat, 160
- rip6s_nosock
 - rip6stat, 160
- rip6s_nosockmcast
 - rip6stat, 160
- rip6s_opackets
 - rip6stat, 161
- rip6stat, 160
- raw_ip6.c, 544
- raw_ip6.h, 545
- rip6s_badsum, 160
- rip6s_fullsock, 160
- rip6s_ipackets, 160
- rip6s_isum, 160
- rip6s_nosock, 160
- rip6s_nosockmcast, 160
- rip6s_opackets, 161
- rip_recvspace
 - raw_ip6.c, 544
- rip_sendspace
 - raw_ip6.c, 544
- ripcb
 - icmp6.c, 244
 - raw_ip6.c, 544
- ripcbinfo
 - icmp6.c, 244
 - in6_ifattach.c, 298
 - raw_ip6.c, 544
- RIPEMD160_RESULTLEN
 - ah_core.c, 187
- RIPV6RCVQ
 - in6_proto.c, 315
- RIPV6SNDQ
 - in6_proto.c, 316
- rnh
 - mtx_arg, 137
 - rtqk_arg, 163
- ro
 - ipsec_output_state, 115
- route6.c
 - __P, 547
 - ip6_rthdr0, 547
 - route6_input, 547
- route6_input
 - route6.c, 547
- router
 - nd_pfxrouter, 144
- rt6_deleteroute
 - nd6_rtr.c, 533
- rt6_flush
 - nd6_rtr.c, 533
- rt6_key
 - ip6_input.c, 370
- rtaddr
 - in6_defrouter, 43
 - in6_drlist, 45

- nd_defrouter, 139
- rtdetq, 162
 - ifp, 162
 - m, 162
 - next, 162
- RTE_FOUND
 - ip6_mroute.c, 382
- RTF_ANNOUNCE
 - nd6.h, 504
- rtlifetime
 - in6_defrouter, 43
 - in6_drlist, 46
 - nd_defrouter, 139
- rtpref
 - nd6_rtr.c, 533
- RTPREF_HIGH
 - nd6_rtr.c, 522
- RTPREF_INVALID
 - nd6_rtr.c, 522
- RTPREF_LOW
 - nd6_rtr.c, 522
- RTPREF_MEDIUM
 - nd6_rtr.c, 522
- RTPREF_RESERVED
 - nd6_rtr.c, 522
- RTPRF_OURS
 - in6_rmx.c, 326
- rtq_minreallyold
 - in6_rmx.c, 329
- rtq_mtutimer
 - in6_rmx.c, 329
- rtq_reallyold
 - in6_rmx.c, 329
- RTQ_TIMEOUT
 - in6_rmx.c, 326
- rtq_timeout
 - in6_rmx.c, 329
- rtq_timer
 - in6_rmx.c, 329
- rtq_toomany
 - in6_rmx.c, 329
- rtqk_arg, 163
 - draining, 163
 - found, 163
 - killed, 163
 - mode, 163
 - nextstop, 163
 - rn timer, 163
 - updating, 163
- RTR_SOLICITATION_INTERVAL
 - nd6.h, 504
- RTRADDR
 - nd6.c, 481
- ru_a
 - randomtab, 157
- ru_agen
 - randomtab, 157
- ru_b
 - randomtab, 157
- ru_bits
 - randomtab, 157
- ru_counter
 - randomtab, 158
- ru_g
 - randomtab, 158
- ru_gen
 - randomtab, 158
- ru_m
 - randomtab, 158
- ru_max
 - randomtab, 158
- ru_msb
 - randomtab, 158
- ru_n
 - randomtab, 158
- ru_out
 - randomtab, 158
- ru_reseed
 - randomtab, 158
- ru_seed
 - randomtab, 159
- ru_seed2
 - randomtab, 159
- ru_x
 - randomtab, 159
- s6_addr
 - in6.h, 279
- s6_addr16
 - in6.h, 279
- s6_addr32
 - in6.h, 279
- s6_addr8
 - in6.h, 279
- s6id_list
 - scope6_id, 165
- sa6_any
 - in6.c, 261
 - in6.h, 281
- sa6_embedscope
 - scope6.c, 551
- sa6_recoverscope
 - scope6.c, 552
- saidx
 - ipsecrequest, 117
- satosin6
 - in6.h, 279
 - in6_pcb.h, 308

- raw_ip6.c, 539
- sav
 - ah_algorithm_state, 24
 - ipsecrequest, 117
- scope6.c
 - in6_addrscope, 551
 - in6_clearscope, 551
 - in6_setscope, 551
 - ip6_use_defzone, 553
 - sa6_embedscope, 551
 - sa6_recoverscope, 552
 - scope6_addr2default, 552
 - scope6_get, 552
 - scope6_get_default, 552
 - scope6_ifattach, 553
 - scope6_ifdetach, 553
 - scope6_init, 553
 - SCOPE6_LOCK, 550
 - scope6_lock, 553
 - SCOPE6_LOCK_ASSERT, 550
 - SCOPE6_LOCK_INIT, 550
 - scope6_set, 553
 - scope6_setdefault, 553
 - SCOPE6_UNLOCK, 550
 - SID, 550
 - sid_default, 554
- scope6_addr2default
 - scope6.c, 552
- scope6_get
 - scope6.c, 552
- scope6_get_default
 - scope6.c, 552
- scope6_id, 165
 - in6_ifextra, 50
 - s6id_list, 165
- scope6_ifattach
 - scope6.c, 553
- scope6_ifdetach
 - scope6.c, 553
- scope6_init
 - scope6.c, 553
- SCOPE6_LOCK
 - scope6.c, 550
- scope6_lock
 - scope6.c, 553
- SCOPE6_LOCK_ASSERT
 - scope6.c, 550
- SCOPE6_LOCK_INIT
 - scope6.c, 550
- scope6_set
 - scope6.c, 553
- scope6_setdefault
 - scope6.c, 553
- SCOPE6_UNLOCK
 - scope6.c, 550
- scope6_var.h
 - __P, 556
- sctp6_abort
 - sctp6_usrreq.c, 559
- sctp6_attach
 - sctp6_usrreq.c, 559
- sctp6_bind
 - sctp6_usrreq.c, 559
- sctp6_close
 - sctp6_usrreq.c, 559
- sctp6_connect
 - sctp6_usrreq.c, 559
- sctp6_ctlinput
 - sctp6_usrreq.c, 560
- sctp6_disconnect
 - sctp6_usrreq.c, 560
- sctp6_getaddr
 - sctp6_usrreq.c, 560
- sctp6_getcred
 - sctp6_usrreq.c, 560
- sctp6_getpeeraddr
 - sctp6_usrreq.c, 560
- sctp6_in6getaddr
 - sctp6_usrreq.c, 561
- sctp6_input
 - sctp6_usrreq.c, 561
- sctp6_notify_mbuf
 - sctp6_usrreq.c, 561
- sctp6_peeraddr
 - sctp6_usrreq.c, 561
- sctp6_send
 - sctp6_usrreq.c, 561
- sctp6_usrreq.c
 - __FBSDID, 559
 - inetsw, 562
 - sctp6_abort, 559
 - sctp6_attach, 559
 - sctp6_bind, 559
 - sctp6_close, 559
 - sctp6_connect, 559
 - sctp6_ctlinput, 560
 - sctp6_disconnect, 560
 - sctp6_getaddr, 560
 - sctp6_getcred, 560
 - sctp6_getpeeraddr, 560
 - sctp6_in6getaddr, 561
 - sctp6_input, 561
 - sctp6_notify_mbuf, 561
 - sctp6_peeraddr, 561
 - sctp6_send, 561
 - sctp6_usrreqs, 562
 - sctp_no_csum_on_loopback, 562
 - sctp_sendm, 562

- SYSCTL_PROC, 562
- sctp6_usrreqs
 - sctp6_usrreq.c, 562
 - sctp6_var.h, 563
- sctp6_var.h
 - __FBSDID, 563
 - __P, 563
 - sctp6_usrreqs, 563
 - SYSCTL_DECL, 563
- sctp_no_csum_on_loopback
 - sctp6_usrreq.c, 562
- sctp_sendm
 - sctp6_usrreq.c, 562
- SDL
 - nd6.c, 481
 - nd6_nbr.c, 511
 - nd6_rtr.c, 522
- search
 - nd_opts, 143
- secpolicy, 166
 - created, 167
 - dir, 167
 - id, 167
 - lastused, 167
 - lifetime, 167
 - LIST_ENTRY, 167
 - persist, 167
 - policy, 167
 - readonly, 167
 - refcnt, 167
 - req, 168
 - so, 168
 - spidx, 168
 - state, 168
 - TAILQ_ENTRY, 167
 - validtime, 168
- secpolicyindex, 169
 - dst, 169
 - prefd, 169
 - prefs, 169
 - src, 169
 - ul_proto, 169
- secspacq, 170
 - count, 170
 - created, 170
 - LIST_ENTRY, 170
 - spidx, 170
- selectroute
 - in6_src.c, 338
- senderr
 - nd6.c, 482
- set_pim6
 - ip6_mroute.c, 389
- SID
 - scope6.c, 550
- sid_default
 - scope6.c, 554
- SIN6
 - nd6.c, 482
 - nd6_rtr.c, 522
- sin6
 - ip6_mroute.c, 391
- sin6_addr
 - sockaddr_in6, 174
- sin6_family
 - sockaddr_in6, 174
- sin6_flowinfo
 - sockaddr_in6, 174
- sin6_len
 - sockaddr_in6, 175
- sin6_port
 - sockaddr_in6, 175
- sin6_scope_id
 - sockaddr_in6, 175
- sin6tos
 - in6.h, 279
 - in6_pcb.h, 309
- sioc_mif_req6, 171
 - ibytes, 171
 - icount, 171
 - mifi, 171
 - obytes, 171
 - ocount, 171
- sioc_sg_req6, 172
 - bytecnt, 172
 - grp, 172
 - pktcnt, 172
 - src, 172
 - wrong_if, 172
- SIOCAADDRCTL_POLICY
 - in6_var.h, 349
- SIOCAIFADDR_IN6
 - in6_var.h, 349
- SIOCAIFPREFIX_IN6
 - in6_var.h, 349
- SIOCCIFPREFIX_IN6
 - in6_var.h, 350
- SIOCDADDRCTL_POLICY
 - in6_var.h, 350
- SIOCDIFADDR_IN6
 - in6_var.h, 350
- SIOCDIFPREFIX_IN6
 - in6_var.h, 350
- SIOCGDEFIFACE_IN6
 - in6_var.h, 350
- SIOCGDRLST_IN6
 - in6_var.h, 350
- SIOCGETMIFCNT_IN6

- in6_var.h, 350
- SIOCGETSGCNT_IN6
 - in6_var.h, 351
- SIOCGIFADDR_IN6
 - in6_var.h, 351
- SIOCGIFAFLAG_IN6
 - in6_var.h, 351
- SIOCGIFALIFETIME_IN6
 - in6_var.h, 351
- SIOCGIFDSTADDR_IN6
 - in6_var.h, 351
- SIOCGIFINFO_IN6
 - in6_var.h, 351
- SIOCGIFNETMASK_IN6
 - in6_var.h, 351
- SIOCGIFPDSTADDR_IN6
 - in6_var.h, 351
- SIOCGIFPREFIX_IN6
 - in6_var.h, 352
- SIOCGIFPSRCADDR_IN6
 - in6_var.h, 352
- SIOCGIFSTAT_ICMP6
 - in6_var.h, 352
- SIOCGIFSTAT_IN6
 - in6_var.h, 352
- SIOCGNBRINFO_IN6
 - in6_var.h, 352
- SIOCGPRLST_IN6
 - in6_var.h, 352
- SIOCGSCOPE6
 - in6_var.h, 352
- SIOCGSCOPE6DEF
 - in6_var.h, 352
- SIOCSDEFIFACE_IN6
 - in6_var.h, 352
- SIOCSGIFPREFIX_IN6
 - in6_var.h, 353
- SIOCSIFADDR_IN6
 - in6_var.h, 353
- SIOCSIFALIFETIME_IN6
 - in6_var.h, 353
- SIOCSIFDSTADDR_IN6
 - in6_var.h, 353
- SIOCSIFINFO_FLAGS
 - in6_var.h, 353
- SIOCSIFINFO_IN6
 - in6_var.h, 353
- SIOCSIFNETMASK_IN6
 - in6_var.h, 353
- SIOCSIFPHYADDR_IN6
 - in6_var.h, 353
- SIOCSIFPREFIX_IN6
 - in6_var.h, 354
- SIOCSNDFLUSH_IN6
 - in6_var.h, 354
- SIOCSPFXFLUSH_IN6
 - in6_var.h, 354
- SIOCSRTRFLUSH_IN6
 - in6_var.h, 354
- SIOCSSCOPE6
 - in6_var.h, 354
- size_t
 - in6.h, 279
- so
 - secpolicy, 168
- sockaddr_in6, 174
 - sin6_addr, 174
 - sin6_family, 174
 - sin6_flowinfo, 174
 - sin6_len, 175
 - sin6_port, 175
 - sin6_scope_id, 175
- socket_send
 - ip6_mroute.c, 389
- socklen_t
 - in6.h, 279
- sp
 - ipsecrequest, 117
- sp_cachegen
 - ipsec.c, 455
- sp_in
 - inpcbpolicy, 86
- sp_out
 - inpcbpolicy, 86
- spdcachelookup
 - ipsecstat, 122
- spdcachemiss
 - ipsecstat, 122
- spidx
 - secpolicy, 168
 - secspacq, 170
- src
 - secpolicyindex, 169
 - sioc_sg_req6, 172
- src_lladdr
 - nd_opts, 143
- state
 - in6_nbrinfo, 62
 - secpolicy, 168
- SYSCTL_DECL
 - in6_rmx.c, 328
 - in6_src.c, 339
 - ipsec.c, 453
 - sctp6_var.h, 563
 - udp6_var.h, 577
- SYSCTL_INT
 - in6_proto.c, 318
 - in6_rmx.c, 329

- ipsec.c, 453
- nd6.c, 492
- sysctl_ip6_temppltime
 - in6_proto.c, 318
- sysctl_ip6_tempvltime
 - in6_proto.c, 319
- SYSCTL_NODE
 - in6_proto.c, 319
 - in6_src.c, 339
 - nd6.c, 492
- SYSCTL_OID
 - in6_proto.c, 319
- SYSCTL_PROC
 - sctp6_usrreq.c, 562
 - udp6_usrreq.c, 572
- SYSCTL_STRING
 - in6_proto.c, 319
- SYSCTL_STRUCT
 - in6_proto.c, 319
 - ipsec.c, 453
- SYSINIT
 - ip6_input.c, 375
- TAILQ_ENTRY
 - nd_defrouter, 138
 - secpolicy, 167
- TAILQ_HEAD
 - in6_src.c, 339
 - nd6.h, 506
 - nd6_nbr.c, 518
- tcp6_usrreqs
 - tcp6_var.h, 564
- tcp6_var.h
 - __P, 564
 - tcp6_usrreqs, 564
 - tcp_rtlookup6, 564
 - tcp_v6mssdflt, 564
- tcp_rtlookup6
 - tcp6_var.h, 564
- tcp_v6mssdflt
 - tcp6_var.h, 564
- TEMPADDR_REGEN_ADVANCE
 - nd6.h, 504
- tgt_lladdr
 - nd_opts, 143
- TUNABLE_INT
 - in6_proto.c, 319
- tunifp
 - ipsecrequest, 118
- TV_DELTA
 - ip6_mroute.c, 382
- TV_LT
 - ip6_mroute.c, 383
- TWOHOUR
 - nd6_rtr.c, 522
- udbinfo
 - in6_ifattach.c, 298
- udp6_abort
 - udp6_usrreq.c, 572
- udp6_append
 - udp6_usrreq.c, 572
- udp6_attach
 - udp6_usrreq.c, 573
- udp6_bind
 - udp6_usrreq.c, 573
- udp6_close
 - udp6_usrreq.c, 573
- udp6_connect
 - udp6_usrreq.c, 573
- udp6_ctlinput
 - udp6_usrreq.c, 574
- udp6_detach
 - udp6_usrreq.c, 574
- udp6_disconnect
 - udp6_usrreq.c, 574
- udp6_getcred
 - udp6_usrreq.c, 575
- udp6_input
 - udp6_usrreq.c, 575
- udp6_output
 - udp6_output.c, 567
- udp6_output.c
 - in6pcb, 567
 - udp6_output, 567
 - udp6s_opackets, 567
 - udp6stat, 567
- udp6_recvspace
 - in6_proto.c, 323
- udp6_send
 - udp6_usrreq.c, 575
- udp6_sendspace
 - in6_proto.c, 323
- udp6_usrreq.c
 - __P, 572
 - inetsw, 576
 - SYSCTL_PROC, 572
 - udp6_abort, 572
 - udp6_append, 572
 - udp6_attach, 573
 - udp6_bind, 573
 - udp6_close, 573
 - udp6_connect, 573
 - udp6_ctlinput, 574
 - udp6_detach, 574
 - udp6_disconnect, 574
 - udp6_getcred, 575
 - udp6_input, 575

- udp6_send, 575
- udp6_usrreqs, 576
- udp6_usrreqs
 - udp6_usrreq.c, 576
 - udp6_var.h, 577
- udp6_var.h
 - __P, 577
 - SYSCTL_DECL, 577
 - udp6_usrreqs, 577
- udp6s_opackets
 - udp6_output.c, 567
- udp6stat
 - udp6_output.c, 567
- ul_proto
 - secpolicyindex, 169
- unused1
 - omrt6msg, 152
- unused2
 - omrt6msg, 153
- UPCALL_EXPIRE
 - ip6_mroute.c, 383
- updating
 - rtqk_arg, 163
- use
 - in6_addrpolicy, 40
- v
 - cblock, 25
- validtime
 - secpolicy, 168
- vlttime
 - in6_oprlist, 68
 - in6_prefix, 71
 - in6_prlist, 79
- vshiffl
 - ipsecc, 453
- w_req
 - walkarg, 176
- walk_addrsel_policy
 - in6_src.c, 339
- walkarg, 176
 - w_req, 176
- wrong_if
 - sioc_sg_req6, 172
- zero
 - nd_opts, 143
- zeroin6_addr
 - in6_pcb.c, 307
 - in6_var.h, 357