NAME

icmp6 - Internet Control Message Protocol for IPv6

SYNOPSIS

#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/icmp6.h>

int
socket(*AF_INET6*, *SOCK_RAW*, *IPPROTO_ICMPV6*);

DESCRIPTION

ICMPv6 is the error and control message protocol used by IPv6 and the IPv6 protocol family (see ip6(4) and inet6(4)). It may be accessed through a "raw socket" for network monitoring and diagnostic functions.

The *proto* parameter to the socket(2) call to create an ICMPv6 socket may be obtained from getprotobyname(3). ICMPv6 sockets are connectionless, and are normally used with the sendto(2) and recvfrom(2) calls, though the connect(2) call may also be used to fix the destination for future packets (in which case read(2) or recv(2) and write(2) or send(2) system calls may be used).

Outgoing packets automatically have an IPv6 header prepended to them (based on the destination address). Incoming packets on the socket are received with the IPv6 header and any extension headers removed.

Types

ICMPv6 messages are classified according to the type and code fields present in the ICMPv6 header. The abbreviations for the types and codes may be used in rules in pf.conf(5). The following types are defined:

Num	Abbrev.	Description
1	unreach	Destination unreachable
2	toobig	Packet too big
3	timex	Time exceeded
4	paramprob	Invalid IPv6 header
128	echoreq	Echo service request
129	echorep	Echo service reply

130
groupqry Group membership query
130
listqry Multicast listener query
131
grouprep Group membership report
131
listenrep Multicast listener report
132
groupterm Group membership termination
132
listendone Multicast listener done
133
routersol Router solicitation
134
routeradv Router advertisement
135
neighbrsol Neighbor solicitation
136
neighbradv Neighbor advertisement
137
redir Shorter route exists
138
routrrenum Route renumbering
139
fqdnreq FQDN query
139
niqry Node information query
139
wrureq Who-are-you request
140
fqdnrep FQDN reply
140
nirep Node information reply
140
wrurep Who-are-you reply
200
mtraceresp mtrace response
201
mtrace mtrace messages

The following codes are defined:

Num	Abbrev.	Type	Description
0	noroute-unr	unreach	No route to destination
1	admin-unr	unreach	Administratively prohibited
2	beyond-unr	unreach	Beyond scope of source address
2	notnbr-unr	unreach	Not a neighbor (obsolete)
3	addr-unr	unreach	Address unreachable
4	port-unr	unreach	Port unreachable
0	transit	timex	Time exceeded in transit
1	reassemb	timex	Time exceeded in reassembly
0	badhead	paramprob	Erroneous header field
1	nxthdr	paramprob	Unrecognized next header
2		redir	Unrecognized option
0	redironlink	redir	Redirection to on-link node
1	redirrouter	redir	Redirection to better router

Headers

All ICMPv6 messages are prefixed with an ICMPv6 header. This header corresponds to the *icmp6_hdr* structure and has the following definition:

```
struct icmp6_hdr {
        uint8_t  icmp6_type;    /* type field */
        uint8_t  icmp6_code;    /* code field */
        uint16_t icmp6_cksum;   /* checksum field */
        union {
                uint32_t icmp6_un_data32[1]; /* type-specific */
                uint16_t icmp6_un_data16[2]; /* type-specific */
                uint8_t  icmp6_un_data8[4];  /* type-specific */
        } icmp6_dataun;
} __packed;

#define icmp6_data32        icmp6_dataun.icmp6_un_data32
#define icmp6_data16        icmp6_dataun.icmp6_un_data16
#define icmp6_data8         icmp6_dataun.icmp6_un_data8
#define icmp6_pptr icmp6_data32[0]    /* parameter prob */
#define icmp6_mtu icmp6_data32[0]     /* packet too big */
#define icmp6_id   icmp6_data16[0]    /* echo request/reply */
#define icmp6_seq  icmp6_data16[1]    /* echo request/reply */
```

```
#define icmp6_maxdelay  icmp6_data16[0]  /* mcast group membership*/
```

icmp6_type describes the type of the message. Suitable values are defined in <*netinet/icmp6.h*>. *icmp6_code* describes the sub-type of the message and depends on *icmp6_type*. *icmp6_cksum* contains the checksum for the message and is filled in by the kernel on outgoing messages. The other fields are used for type-specific purposes.

Filters

Because of the extra functionality of ICMPv6 in comparison to ICMPv4, a larger number of messages may be potentially received on an ICMPv6 socket. Input filters may therefore be used to restrict input to a subset of the incoming ICMPv6 messages so only interesting messages are returned by the recv(2) family of calls to an application.

The *icmp6_filter* structure may be used to refine the input message set according to the ICMPv6 type. By default, all messages types are allowed on newly created raw ICMPv6 sockets. The following macros may be used to refine the input set:

void **ICMP6_FILTER_SETPASSALL**(*struct icmp6_filter *filterp*)
Allow all incoming messages. *filterp* is modified to allow all message types.

void **ICMP6_FILTER_SETBLOCKALL**(*struct icmp6_filter *filterp*)
Ignore all incoming messages. *filterp* is modified to ignore all message types.

void **ICMP6_FILTER_SETPASS**(*int type, struct icmp6_filter *filterp*)
Allow ICMPv6 messages with the given *type*. *filterp* is modified to allow such messages.

void **ICMP6_FILTER_SETBLOCK**(*int type, struct icmp6_filter *filterp*)
Ignore ICMPv6 messages with the given *type*. *filterp* is modified to ignore such messages.

int **ICMP6_FILTER_WILLPASS**(*int type, const struct icmp6_filter *filterp*)
Determine if the given filter will allow an ICMPv6 message of the given type.

int **ICMP6_FILTER_WILLBLOCK**(*int type, const struct icmp6_filter *filterp*)
Determine if the given filter will ignore an ICMPv6 message of the given type.

The getsockopt(2) and setsockopt(2) calls may be used to obtain and install the filter on ICMPv6 sockets at option level IPPROTO_ICMPV6 and name ICMP6_FILTER with a pointer to the *icmp6_filter* structure as the option value.

SEE ALSO

getsockopt(2), recv(2), send(2), setsockopt(2), socket(2), getprotobyname(3), inet6(4), ip6(4), netintro(4)

W. Stevens and M. Thomas, *Advanced Sockets API for IPv6*, RFC 2292, February 1998.

A. Conta and S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 2463, December 1998.

W. Stevens, M. Thomas, E. Nordmark, and T. Jinmei, *Advanced Sockets Application Program Interface (API) for IPv6*, RFC 3542, May 2003.

A. Conta, S. Deering, and M. Gupta, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443, March 2006.