Internet Draft   A. Conta, Transwitch
IPv6 Working Group   S. Deering, Cisco Systems
21 **October 2004**   M. Gupta, Nokia (ed.)

                   Internet Control Message Protocol (ICMPv6)
                   for the Internet Protocol Version 6 (IPv6)
                                Specification

                      <draft-ietf-ipngwg-icmp-v3-05.txt>

Status of this Memo

Abstract

This document describes the format of a set of control messages used
   in ICMPv6 (Internet Control Message Protocol). ICMPv6 is the
   Internet Control Message Protocol for Internet Protocol version 6
   (IPv6).

Table of Contents

1. Introduction

The Internet Protocol, version 6 (IPv6) is a new version of IP. IPv6
uses the Internet Control Message Protocol (ICMP) as defined for IPv4
[RFC-792], with a number of changes. The resulting protocol is
called ICMPv6, and has an IPv6 Next Header value of 58.

This document describes the format of a set of control messages used
in ICMPv6. It does not describe the procedures for using these
messages to achieve functions like Path MTU discovery; such
procedures are described in other documents (e.g., [PMTU]). Other
documents may also introduce additional ICMPv6 message types, such as
Neighbor Discovery messages [IPv6-DISC], subject to the general rules
for ICMPv6 messages given in section 2 of this document.

Terminology defined in the IPv6 specification [IPv6] and the IPv6
Routing and Addressing specification [IPv6-ADDR] applies to this
document as well.

This document obsoletes RFC 2463 [RFC-2463].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC-2119].

2. ICMPv6 (ICMP for IPv6)

ICMPv6 is used by IPv6 nodes to report errors encountered in
processing packets, and to perform other internet-layer functions,
such as diagnostics (ICMPv6 "ping"). ICMPv6 is an integral part of
IPv6 and MUST be fully implemented by every IPv6 node.

2.1 Message General Format

ICMPv6 messages are grouped into two classes: error messages and
informational messages. Error messages are identified as such by
having a zero in the high-order bit of their message Type field

values. Thus, error messages have message Types from 0 to 127;
informational messages have message Types from 128 to 255.

This document defines the message formats for the following ICMPv6
messages:

ICMPv6 error messages:

    1   Destination Unreachable   (see section 3.1)
    2   Packet Too Big   (see section 3.2)
    3   Time Exceeded   (see section 3.3)
    4   Parameter Problem   (see section 3.4)

    100 Private experimentation
    101 Private experimentation

    127 Reserved for expansion of ICMPv6 error messages

ICMPv6 informational messages:

    128 Echo Request   (see section 4.1)
    129 Echo Reply   (see section 4.2)

    200 Private experimentation
    201 Private experimentation

    255 Reserved for expansion of ICMPv6 informational messages

Type values 100, 101, 200, and 201 are reserved for private
experimentation. These are not intended for general use. It is
expected that multiple concurrent experiments will be done with the
same type values. Any wide scale and/or uncontrolled usage should
obtain real allocations as defined in section 6.

Type value 255 is reserved for future expansion of the type value
range if there should be a shortage in the future. The details of
this are left for future work. One possible way of doing this that
would not cause any problems with current implementations is if the
type equals 255, use the code field for the new assignment. Existing
implementations would ignore the new assignments as specified in
section 2.4, section (b). The new messages using these expanded type
values, could assign fields in the message body for it's code values.

Every ICMPv6 message is preceded by an IPv6 header and zero or more
IPv6 extension headers. The ICMPv6 header is identified by a Next
Header value of 58 in the immediately preceding header. (NOTE: this
is different than the value used to identify ICMP for IPv4.)

The ICMPv6 messages have the following general format:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +                         Message Body                          +
 |                                                               |
```

The type field indicates the type of the message. Its value
determines the format of the remaining data.

The code field depends on the message type. It is used to create an
additional level of message granularity.

The checksum field is used to detect data corruption in the ICMPv6
message and parts of the IPv6 header.

The subclass of ICMPv6 messages used for reporting errors, i.e.,
those with a Type value between 0 and 127, inclusive, all have the
following, more specific format:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   type-specific data (32 bits)                |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   As much of invoking packet                  |
 +                as possible without the ICMPv6 packet          +
 |                exceeding the minimum IPv6 MTU [IPv6]          |
```

2.2 Message Source Address Determination

A node that sends an ICMPv6 message has to determine both the Source
and Destination IPv6 Addresses in the IPv6 header before calculating

the checksum. If the node has more than one unicast address, it MUST
choose the Source Address of the message as follows:

  (a) If the message is a response to a message sent to one of the
      node's unicast addresses, the Source Address of the reply MUST
      be that same address.

(b) If the message is a response to a message sent to a multicast or
    anycast group in which the node is a member, the Source Address
    of the reply MUST be a unicast address belonging to the
    interface on which the multicast or anycast packet was received.

(c) If the message is a response to a message sent to an address
    that does not belong to the node, the Source Address SHOULD be
    that unicast address belonging to the node that will be most
    helpful in diagnosing the error. For example, if the message is
    a response to a packet forwarding action that cannot complete
    successfully, the Source Address SHOULD be a unicast address
    belonging to the interface on which the packet forwarding
    failed.

(d) Otherwise, the node's routing table must be examined to
    determine which interface will be used to transmit the message
    to its destination, and a unicast address belonging to that
    interface MUST be used as the Source Address of the message.

2.3 Message Checksum Calculation

The checksum is the 16-bit one's complement of the one's complement
sum of the entire ICMPv6 message starting with the ICMPv6 message
type field, prepended with a "pseudo-header" of IPv6 header fields,
as specified in [IPv6, section 8.1]. The Next Header value used in
the pseudo-header is 58. (NOTE: the inclusion of a pseudo-header in
the ICMPv6 checksum is a change from IPv4; see [IPv6] for the
rationale for this change.)

For computing the checksum, the checksum field is first set to zero.

2.4 Message Processing Rules

Implementations MUST observe the following rules when processing
ICMPv6 messages (from [RFC-1122]):

(a) If an ICMPv6 error message of unknown type is received, it MUST
    be passed to the upper layer.

(b) If an ICMPv6 informational message of unknown type is received,
        it MUST be silently discarded.

    (c) Every ICMPv6 error message (type < 128) MUST include as much of
        the IPv6 offending (invoking) packet (the packet that caused the
        error) as possible without making the error message packet
        exceed the minimum IPv6 MTU [IPv6].

(d) In those cases where the internet-layer protocol is required to
    pass an ICMPv6 error message to the upper-layer process, the
    upper-layer protocol type is extracted from the original packet
    (contained in the body of the ICMPv6 error message) and used to
    select the appropriate upper-layer process to handle the error.

    If the original packet had an unusually large amount of
    extension headers, it is possible that the upper-layer protocol
    type may not be present in the ICMPv6 message, due to truncation
    of the original packet to meet the minimum IPv6 MTU [IPv6]
    limit. In that case, the error message is silently dropped
    after any IPv6-layer processing.

(e) An ICMPv6 error message MUST NOT be sent as a result of
    receiving:

    (e.1) an ICMPv6 error message, or

    (e.2) an ICMPv6 redirect message [IPv6-DISC], or

    (e.3) a packet destined to an IPv6 multicast address (there are
          two exceptions to this rule: (1) the Packet Too Big
          Message - Section 3.2 - to allow Path MTU discovery to
          work for IPv6 multicast, and (2) the Parameter Problem
          Message, Code 2 - Section 3.4 - reporting an unrecognized
          IPv6 option that has the Option Type highest-order two
          bits set to 10), or

    (e.4) a packet sent as a link-layer multicast, (the exception
          from e.3 applies to this case too), or

    (e.5) a packet sent as a link-layer broadcast, (the exception
          from e.3 applies to this case too), or

    (e.6) a packet whose source address does not uniquely identify
          a single node -- e.g., the IPv6 Unspecified Address, an
          IPv6 multicast address, or an address known by the ICMP
          message sender to be an IPv6 anycast address.

(f) Finally, in order to limit the bandwidth and forwarding costs

incurred by sending ICMPv6 error messages, an IPv6 node MUST
limit the rate of ICMPv6 error messages it sends. This
situation may occur when a source sending a stream of erroneous
packets fails to heed the resulting ICMPv6 error messages.

A recommended method for implementing the rate-limiting function
is a token bucket, limiting the average rate of transmission to
N, where N can either be packets/second or a fraction of the

attached link's bandwidth, but allowing up to B error messages
to be transmitted in a burst, as long as the long-term average
is not exceeded.

Rate-limiting mechanisms which cannot cope with bursty traffic
(e.g., traceroute) are not recommended; for example a simple
timer-based implementation, allowing an error message every T
milliseconds (even with low values for T), is not reasonable.

The rate-limiting parameters SHOULD be configurable. In the
case of a token-bucket implementation, the best defaults depend
on where the implementation is expected to be deployed (e.g., a
high-end router vs. an embedded host). For example, in a
small/mid -sized device, the possible defaults could be B=10,
N=10/s.

NOTE: THE RESTRICTIONS UNDER (e) AND (f) ABOVE TAKE PRECEDENCE OVER
ANY REQUIREMENT ELSEWHERE IN THIS DOCUMENT FOR SENDING ICMP ERROR
MESSAGES.

The following sections describe the message formats for the above
ICMPv6 messages.

3. ICMPv6 Error Messages

3.1 Destination Unreachable Message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Unused                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    As much of invoking packet                |
+                as possible without the ICMPv6 packet         +
|                exceeding the minimum IPv6 MTU [IPv6]         |
```

IPv6 Fields:

Destination Address

                Copied from the Source Address field of the invoking
                packet.

ICMPv6 Fields:

Type   1

Code   0 - no route to destination
              1 - communication with destination
                    administratively prohibited
              2 - beyond scope of source address
              3 - address unreachable
              4 - port unreachable
              5 - source address failed ingress/egress policy
              6 - reject route to destination

Unused   This field is unused for all code values.
              It must be initialized to zero by the sender
              and ignored by the receiver.
Description

A Destination Unreachable message SHOULD be generated by a router, or
by the IPv6 layer in the originating node, in response to a packet
that cannot be delivered to its destination address for reasons other
than congestion. (An ICMPv6 message MUST NOT be generated if a
packet is dropped due to congestion.)

If the reason for the failure to deliver is lack of a matching entry
in the forwarding node's routing table, the Code field is set to 0

draft-ietf-ipngwg-icmp-v3-05.txt   [Page 9]

(NOTE: this error can occur only in nodes that do not hold a "default route" in their routing tables).

If the reason for the failure to deliver is administrative prohibition, e.g., a "firewall filter", the Code field is set to 1.

If the reason for the failure to deliver is that the destination is beyond the scope of the source address, the Code field is set to 2. This condition can occur only when the scope of the source address is smaller than the scope of the destination address (e.g., when a packet has a link-local source address and a global-scope destination address) and the packet cannot be delivered to the destination without leaving the scope of the source address.

If the reason for the failure to deliver can not be mapped to any of other codes, the Code field is set to 3. The example of such cases are inability to resolve the IPv6 destination address into a corresponding link address, or a link-specific problem of some sort.

One specific case in which a Destination Unreachable message with a code 3 is sent is in response to a packet received by a router from a point-to-point link, destined to an address within a subnet assigned to that same link (other than one of the receiving router's own addresses). In such a case, the packet MUST NOT be forwarded back onto the arrival link.

A destination node SHOULD send a Destination Unreachable message with Code 4 in response to a packet for which the transport protocol (e.g., UDP) has no listener, if that transport protocol has no alternative means to inform the sender.

If the reason for the failure to deliver is that packet with this source address is not allowed due to ingress or egress filtering policies, the Code field is set to 5.

If the reason for the failure to deliver is that the route to the destination is a reject route, the Code field is set to 6. This may occur if the router has been configured to reject all the traffic for a specific prefix.

Codes 5 and 6 are more informative subsets of code 1.

Upper layer notification

A node receiving the ICMPv6 Destination Unreachable message MUST
notify the upper-layer process.

3.2 **Packet Too Big Message**

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |     Code      |          Checksum             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             MTU                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    As much of invoking packet                 |
   +                as possible without the ICMPv6 packet          +
   |                exceeding the minimum IPv6 MTU [IPv6]          |
```

IPv6 Fields:

Destination Address

            Copied from the Source Address field of the invoking
            packet.

ICMPv6 Fields:

Type   2

Code   Set to 0 (zero) by the sender and ignored by the
                receiver

MTU   The Maximum Transmission Unit of the next-hop link.

Description

A Packet Too Big MUST be sent by a router in response to a packet
that it cannot forward because the packet is larger than the MTU of
the outgoing link. The information in this message is used as part
of the Path MTU Discovery process [PMTU].

Sending a Packet Too Big Message makes an exception to one of the
rules of when to send an ICMPv6 error message, in that unlike other

messages, it is sent in response to a packet received with an IPv6
multicast destination address, or a link-layer multicast or link-
layer broadcast address.

Upper layer notification

An incoming Packet Too Big message MUST be passed to the upper-layer
process.

3.3 **Time Exceeded Message**

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |     Code      |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             Unused                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    As much of invoking packet                 |
    +                as possible without the ICMPv6 packet          +
    |                exceeding the minimum IPv6 MTU [IPv6]          |
```

IPv6 Fields:

Destination Address
              Copied from the Source Address field of the invoking
              packet.

ICMPv6 Fields:

Type   3

Code   0 - hop limit exceeded in transit

              1 - fragment reassembly time exceeded

Unused   This field is unused for all code values.
              It must be initialized to zero by the sender
              and ignored by the receiver.

Description

If a router receives a packet with a Hop Limit of zero, or a router
decrements a packet's Hop Limit to zero, it MUST discard the packet
and send an ICMPv6 Time Exceeded message with Code 0 to the source of
the packet. This indicates either a routing loop or too small an
initial Hop Limit value.

An ICMPv6 Time Exceeded message with Code 1 is used to report
fragment reassembly timeout, as specified in [IPv6, Section 4.5].

The rules for selecting the Source Address of this message are
defined in section 2.2.

Upper layer notification

An incoming Time Exceeded message MUST be passed to the upper-layer
process.

3.4 **Parameter Problem Message**


```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Pointer                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    As much of invoking packet                 |
 +                as possible without the ICMPv6 packet          +
 |                exceeding the minimum IPv6 MTU [IPv6]          |
```


IPv6 Fields:

Destination Address

                Copied from the Source Address field of the invoking
                packet.

ICMPv6 Fields:

Type   4

Code   0 - erroneous header field encountered

                1 - unrecognized Next Header type encountered

                2 - unrecognized IPv6 option encountered

Pointer   Identifies the octet offset within the
                invoking packet where the error was detected.

                The pointer will point beyond the end of the ICMPv6
                packet if the field in error is beyond what can fit
                in the maximum size of an ICMPv6 error message.

Description

      If an IPv6 node processing a packet finds a problem with a field in
      the IPv6 header or extension headers such that it cannot complete
      processing the packet, it MUST discard the packet and SHOULD send an
      ICMPv6 Parameter Problem message to the packet's source, indicating
      the type and location of the problem.

      The pointer identifies the octet of the original packet's header
      where the error was detected. For example, an ICMPv6 message with
      Type field = 4, Code field = 1, and Pointer field = 40 would indicate

draft-ietf-ipngwg-icmp-v3-05.txt   [Page 14]

that the IPv6 extension header following the IPv6 header of the
original packet holds an unrecognized Next Header field value.

Upper layer notification

A node receiving this ICMPv6 message MUST notify the upper-layer
process.

4. ICMPv6 Informational Messages

4.1 Echo Request Message


```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |     Code      |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Identifier          |        Sequence Number        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Data ...
    +-+-+-+-+-
```
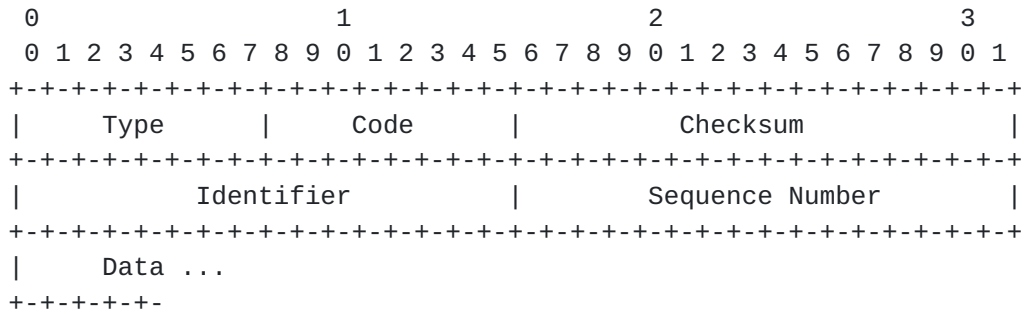

   IPv6 Fields:

   Destination Address

                 Any legal IPv6 address.

   ICMPv6 Fields:

   Type   128

   Code   0

   Identifier   An identifier to aid in matching Echo Replies
                   to this Echo Request. May be zero.

   Sequence Number

                   A sequence number to aid in matching Echo Replies
                   to this Echo Request. May be zero.

   Data   Zero or more octets of arbitrary data.

Description

   Every node MUST implement an ICMPv6 Echo responder function that
   receives Echo Requests and sends corresponding Echo Replies. A node
   SHOULD also implement an application-layer interface for sending Echo
   Requests and receiving Echo Replies, for diagnostic purposes.

   Upper layer notification

   Echo Request messages MAY be passed to processes receiving ICMP
   messages.

4.2 **Echo Reply Message**

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Data ...
+-+-+-+-+-
```

IPv6 Fields:

Destination Address

              Copied from the Source Address field of the invoking
              Echo Request packet.

ICMPv6 Fields:

Type   129

Code   0

Identifier   The identifier from the invoking Echo Request message.

Sequence   The sequence number from the invoking Echo Request
Number   message.

Data   The data from the invoking Echo Request message.

Description

Every node MUST implement an ICMPv6 Echo responder function that
receives Echo Requests and sends corresponding Echo Replies. A node
SHOULD also implement an application-layer interface for sending Echo

Requests and receiving Echo Replies, for diagnostic purposes.

The source address of an Echo Reply sent in response to a unicast
Echo Request message MUST be the same as the destination address of
that Echo Request message.

An Echo Reply SHOULD be sent in response to an Echo Request message
sent to an IPv6 multicast or anycast address. In this case, the
source address of the reply MUST be a unicast address belonging to
the interface on which the Echo Request message was received.

The data received in the ICMPv6 Echo Request message MUST be returned entirely and unmodified in the ICMPv6 Echo Reply message.

Upper layer notification

Echo Reply messages MUST be passed to the process that originated an Echo Request message. An Echo Reply message MAY be passed to processes that did not originate the Echo Request message.

5. Security Considerations

5.1 Authentication and Confidentiality of ICMP messages

ICMP protocol packet exchanges can be authenticated using the IP
Authentication Header [IPv6-AUTH] or IP Encapsulating Security
Payload Header [IPv6-ESP]. Confidentiality for the ICMP protocol
packet exchanges can be achieved using IP Encapsulating Security
Payload Header [IPv6-ESP]. A node SHOULD include an Authentication
Header or Encapsulating Security Payload Header when sending ICMP
messages if a security association for use with the IP Authentication
Header or IP Encapsulating Security Payload Header exists for the
destination address. The security associations may have been created
through manual configuration or through the operation of some key
management protocol.

Received ICMP packets that have Authentication Header or
Encapsulating Security Payload Header must be processed as specified
in [IPv6-AUTH] and [IPv6-ESP]. The ICMP packets that fail the
security processing MUST be ignored and discarded.

The system administrator MAY be allowed to configure a node to ignore
any ICMP messages that are not authenticated using either the
Authentication Header or Encapsulating Security Payload. If
provided, such a switch SHOULD default to allowing unauthenticated
messages. Note that setting up Security Associations to deal with
all the required ICMP packets is a very difficult task (e.g.,
consider the Path MTU Discovery packets). So Path MTU Discovery (and
possibly some others) may not work if the node only allows
authenticated ICMP packets.

5.2 ICMP Attacks

ICMP messages may be subject to various attacks. A complete
discussion can be found in the IP Security Architecture [IPv6-SA]. A
brief discussion of such attacks and their prevention is as follows:

1. ICMP messages may be subject to actions intended to cause the
   receiver to believe the message came from a different source than
   the message originator. The protection against this attack can be
   achieved by applying the IPv6 Authentication mechanism [IPv6-AUTH]

to the ICMP message.

   2. ICMP messages may be subject to actions intended to cause the
      message or the reply to it go to a destination different than the
      message originator's intention. The protection against this
      attack can be achieved by using the Authentication Header
      [IPv6-AUTH] or the Encapsulating Security Payload Header

[IPv6-ESP]. Authentication Header provides the protection against
change for the source and the destination address of the IP
packet. Encapsulating Security Payload Header does not provide
this protection but the ICMP checksum calculation includes the
source and the destination addresses and the Encapsulating
Security Payload Header protects the checksum. Therefore, the
combination of ICMP checksum and the Encapsulating Security
Payload Header provides the protection against this attack. The
protection provided by the Encapsulating Security Payload Header
will not be as strong as the protection provided by the
Authentication Header.

3. ICMP messages may be subject to changes in the message fields, or
   payload. The authentication [IPv6-AUTH] or encryption [IPv6-ESP]
   of the ICMP message is a protection against such actions.

4. ICMP messages may be used as attempts to perform denial of service
   attacks by sending back to back erroneous IP packets. An
   implementation that correctly followed section 2.4, paragraph (f)
   of this specifications, would be protected by the ICMP error rate
   limiting mechanism.

5. The exception number 2 of rule e.3 in section 2.4 gives the
   opportunity to a malicious node to cause a denial of service
   attack to a multicast source. A malicious node can send a
   multicast packet with an unknown destination option marked as
   mandatory with the IPv6 source address of a valid multicast
   source. A large number of destination nodes will send ICMP
   Parameter Problem Message to the multicast source causing a denial
   of service attack. The way multicast traffic is forwarded by the
   multicast routers does require the malicious node to be part of
   the correct multicast path i.e. near to the multicast source.
   This attack can only be avoided by securing the multicast traffic.
   The multicast source should be careful while sending multicast
   traffic with the destination options marked as mandatory because
   they can cause a denial of service attack to themselves if the
   destination option is unknown to a large number of destinations.

6. IANA Considerations

6.1 Procedure for new ICMPV6 Type and Code value assignments

The IPv6 ICMP header [ICMPV6] contains the following fields that
carry values assigned from IANA-managed name spaces: Type and Code.
Code field values are defined relative to a specific Type value.

Values for the IPv6 ICMP Type fields are allocated using the
following procedure:

1. The IANA should allocate and permanently register new ICMPv6 type
   codes from IETF RFC publication. This is for all RFC types
   including standards track, informational, and experimental status
   that originate from the IETF and have been approved by the IESG
   for publication.

2. IETF working groups with working group consensus and area director
   approval can request reclaimable ICMPV6 type code assignments from
   the IANA. The IANA will tag the values as "reclaimable in
   future".

   The "reclaimable in the future" tag will be removed when an RFC is
   published documenting the protocol as defined in 1). This will
   make the assignment permanent and update the reference on the IANA
   web pages.

   At the point where the ICMPv6 type values are 85% assigned, the
   IETF will review the assignments tagged "reclaimable in the
   future" and inform the IANA which ones should be reclaimed and
   reassigned.

3. Requests for new ICMPv6 type value assignments from outside the
   IETF are only made through the publication of an IETF document,
   per 1) above.   Note also that documents published as "RFC Editor
   contributions" [RFC 3667] are not considered to be IETF documents.

   The assignment of new Code values for the Type values defined in this
   document require standards action or IESG approval. The policy for
   assigning Code values for new IPv6 ICMP Types not defined in this

document should be defined in the document defining the new Type
values.

6.2 Assignments for this document

The following should update the assignments located at:

   http://www.iana.org/assignments/icmpv6-parameters

The IANA is requested to reassign ICMPv6 type 1 "Destination
Unreachable" code 2, that was unassigned in [RFC-2463], to:

        2 - beyond scope of source address

The IANA is requested to assign the following two new codes values
for ICMPv6 type 1 "Destination Unreachable":

        5 - source address failed ingress/egress policy
        6 - reject route to destination

The IANA is requested to assign the following new type values:

        100 Private experimentation
        101 Private experimentation

        200 Private experimentation
        201 Private experimentation

        255 Reserved for expansion

7. References

7.1 Normative

    [IPv6]   Deering, S., R. Hinden, "Internet Protocol, Version 6,
                  Specification", RFC2460, December 1998.

    [IPv6-DISC]   Narten, T., E. Nordmark, W. Simpson, "Neighbor Discovery
                  for IP Version 6 (IPv6)", RFC2461, December, 1998.

    [RFC-792]   Postel, J., "Internet Control Message Protocol", STD 5,
                  RFC792, September 1981.

    [RFC-2463]   Conta, A., S. Deering, "Internet Control Message
                  Protocol (ICMPv6) for the Internet Protocol Version 6

(IPv6) Specification", RFC2463, December, 1998.

   [RFC-1122]   Braden, R., "Requirements for Internet Hosts -
                Communication Layers", STD 5, RFC1122, August 1989.

   [RFC-2119]   Bradner, S., "Key words for use in RFCs to Indicate
                Requirement Levels", BCP14, RFC2119, March 1997.

7.2 Informative

   [RFC-2780]   Bradner, S., V. Paxson, "IANA Allocation Guidelines For

                  Values In the Internet Protocol and Related Headers",
                  RFC 2780, March 2000.

   [IPv6-ADDR]   Hinden, R., S. Deering, "IP Version 6 Addressing
                  Architecture", RFC2373, July 1998.

   [PMTU]   McCann, J., S. Deering, J. Mogul, "Path MTU Discovery
                  for IP version 6", RFC1981, August 1996.

   [IPv6-SA]   Kent, S., R. Atkinson, "Security Architecture for the
                  Internet Protocol", RFC1825, November 1998.

   [IPv6-AUTH]   Kent, S., R. Atkinson, "IP Authentication Header", RFC
                  2402, November 1998.

   [IPv6-ESP]   Kent, S., R. Atkinson, "IP Encapsulating Security
                  Payload (ESP)", RFC 2406, November 1998.

8. Acknowledgments

   The document is derived from previous ICMP drafts of the SIPP and
   IPng working group.

   The IPng working group and particularly Robert Elz, Jim Bound, Bill
   Simpson, Thomas Narten, Charlie Lynn, Bill Fink, Scott Bradner,
   Dimitri Haskin, Bob Hinden, Jun-ichiro Itojun Hagino, Tatuya Jinmei,
   Brian Zill, Pekka Savola, and Fred Templin (in chronological order)
   provided extensive review information and feedback.

   Bob Hinden was the document editor for this document.

9. Authors' Addresses

   Alex Conta
   Transwitch Corporation
   3 Enterprise Drive

        Shelton, CT 06484
        USA
        Email: aconta@txc.com

        Stephen Deering
        Cisco Systems, Inc.
        170 West Tasman Drive
        San Jose, CA 95134-1706
        USA

     Mukesh Gupta (ed.)
     Nokia
     313 Fairchild Drive
     Mountain View, CA 94043
     USA
     Phone: +1 650-625-2264
     Email: mukesh.gupta@nokia.com

Appendix A - Changes since RFC 2463

     The following changes were made from RFC 2463:

       - Edited the Abstract to make it a little more elaborate.

       - Corrected typos in section 2.4, where references to sub-bullet e.2
         were supposed to be references to e.3.

       - Removed the Timer-based and the Bandwidth-based methods from the
         example rate-limiting mechanism for ICMP error messages. Added
         Token-bucket based method.

       - Added specification that all ICMP error messages shall have
         exactly 32 bits of type-specific data, so that receivers can
         reliably find the embedded invoking packet even when they don't
         recognize the ICMP message Type.

       - In the description of Destination Unreachable messages, Code 3,
         added rule prohibiting forwarding of packets back onto point-to-
         point links from which they were received, if their destination
         addresses belong to the link itself ("anti-ping-ponging" rule).

       - Added description of Time Exceeded Code 1 (fragment reassembly
         timeout).

       - Added "beyond scope of source address", "source address failed
         ingress/egress policy", and "reject route to destination" messages
         to the family of "unreachable destination" type ICMP error
         messages (section 3.1).

- Reserved some ICMP type values for experimentation.

     - Added a NOTE in section 2.4, that specifies ICMP message
       processing rules precedence.

     - Added ICMP REDIRECT to the list in Section 2.4 e) of cases in
       which ICMP error messages are not to be generated.

draft-ietf-ipngwg-icmp-v3-05.txt   [Page 24]

- Made minor editorial changes in Section 2.3 on checksum
  calculation, and in Section 5.2.

- Clarified in section 4.2, regarding the Echo Reply Message, that
  the source address of an Echo Reply to an anycast Echo Request
  should be a unicast address, as in the case of multicast.

- Revised the Security Considerations section. Added the use of
  Encapsulating Security Payload Header for authentication. Changed
  the requirement of an option of "not allowing unauthenticated ICMP
  messages" to MAY from SHOULD.

- Added a new attack in the list of possible ICMP attacks in section
  5.2.

- Separated References into Normative and Informative.

- Added reference to RFC-2780 "IANA Allocation Guidelines For Values
  In the Internet Protocol and Related Headers"

- Added a procedure for new ICMPv6 Type and Code value assignments
  in the IANA Consideration section.

Full Copyright Statement

Intellectual Property Statement

Acknowledgement