Internet Control Message Protocol (ICMPv6)

ICMPv6 is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions such as diagnostics.

ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node.

ICMPv6 messages are grouped into two classes: error messages and informational messages.

High-order bit of the message Type:
- 0: Error messages (Type: 0 to 127).
- 1: Informational messages (Type: 128 to 255).

Integration of protocols in ICMPv6

ICMPv6 Messages

ICMPv6 error messages:
- 1 Destination Unreachable.
- 2 Packet Too Big.
- 3 Time Exceeded.
- 4 Parameter Problem.

ICMPv6 informational messages:
- 128 Echo Request.
- 129 Echo Reply.

ICMPv6 Messages General Format

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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
</table>

- Type: indicates the type of the message.
- Code: depends on the message type.
- Checksum: is used to detect data corruption in the ICMPv6 message and parts of the IPv6 header.

ICMPv6 Message Processing Rules

- If an ICMPv6 error message of unknown type is received, it must be passed to the upper layer.
- If an ICMPv6 informational message of unknown type is received, it must be silently discarded.
- Every ICMPv6 error message includes as much of the IPv6 offending packet as will fit without making the error message packet exceed the minimum IPv6 MTU.
- In those cases where the internet-layer protocol is required to pass an ICMPv6 message to the upper layer process, the upper-layer protocol type is extracted from the original packet and used to select the appropriate upper-layer process to handle the error.

ICMPv6 Message Processing Rules

- An ICMPv6 error message must not be sent as a result of receiving:
  - An ICMPv6 error message.
  - A packet destined to an IPv6 multicast address*.
  - A packet sent as a link-layer multicast*.
  - A packet sent as a link-layer broadcast*.
  - A packet whose source address does not uniquely identify a single node (e.g. Unspecified Address, Multicast address, Anycast address).

*Exceptions: the Packet Too Big message, Parameter Problem message reporting an unrecognized option with Option type highest-order two bits set to 10.
**ICMPv6 Message Processing Rules**

- In order to limit the bandwidth and forwarding costs incurred sending ICMPv6 error messages, an IPv6 node must limit the rate of ICMPv6 error messages it sends.

  There are a variety of ways of implementing this function, e.g.:

  - Timer-based (limiting the rate of transmission of error messages to a given source or to any source to at most once every T milliseconds).
  - Bandwidth-based (for example, limiting the rate at which error messages are sent from a particular interface to some fraction of the attached link’s bandwidth).

**ICMPv6 Error Message: Destination Unreachable**

- 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Unused</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>-</td>
<td>16 bits</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-</td>
<td>16 bits</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>-</td>
<td>16 bits</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>-</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

  As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU

  IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
  
  Type: 1
  
  Code: 0 – no route to destination
  
  1 – communication with dest prohibited
  
  2 – address unreachable
  
  3 – port unreachable
  
  Unused: Must be initialized to zero by the sender and ignored by the receiver.

**ICMPv6 Error Message: 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

  As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU

  IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
  
  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.

**ICMPv6 Error Message: Parameter Problem**

- If an IPv6 node 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
  
  As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU

  IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
  
  Type: 4
  
  Code: 0 – erroneous header field
  
  1 – unrecognized Next Header type
  
  2 – unrecognized IPv6 option
  
  Pointer: Identifies the octet offset within the invoking packet where the error was detected.

**ICMPv6 Informational Message: Echo Request**

- Every node must implement an ICMPv6 Echo responder function that receives Echo Requests and sends corresponding Echo Replies.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

  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.

  IPv6 Destination Address: Any legal IPv6 address.
  
  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.
ICMPv6 Informational Message: Echo Reply

- Every node must implement an ICMPv6 Echo responder function that receives Echo Requests and sends corresponding Echo Replies. The data received in the ICMPv6 Echo Request message must be returned entirely and unmodified in the ICMPv6 Echo Reply message.

| Field          | Type | Code | Identifier | Sequence Number | Checksum | Data...
|----------------|------|------|------------|-----------------|----------|----------
| IPv6 Destination Address: Copied from the Source Address field of the invoking Echo Request Packet. |
| Type: 129      |      |      |            |                 |          |          
| Code: 0        |      |      |            |                 |          |          
| Identifier: The identifier from the invoking Echo Request message. |
| Sequence Number: The sequence number from the invoking Echo Request message. |
| Data: The data from the invoking Echo Request message. |

ICMPv6: Security Considerations

- ICMP protocol packet exchanges can be authenticated using the IP Authentication Header. A node should include an Authentication Header when sending ICMP messages if a security association for use with the IP Authentication header exists for the destination address.

- Received Authentication Headers in ICMP packets must be verified for correctness and packets with incorrect authentication must be ignored and discarded.

Summary / ICMPv6

ICMPv4 + ARP + IGMP \rightarrow ICMPv6 (ND, MLD)

- Error Messages:
  - Destination Unreachable: Packet cannot be delivered
  - Packet Too Big: Size > MTU
  - Time Exceeded: Hop Limit = 0
  - Parameter Problem: Field Error

- Informational:
  - Echo Request / Reply: Ping
  - More messages for MLD, ND, Mobility

Neighbor Discovery

- Nodes use Neighbor Discovery (ND) to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid.

- Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf.

- Nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses.

- When a router or the path to a router fails, a host actively searches for functioning alternates.

Neighbor Discovery Features

- This protocol solves a set of problems related to the interaction between nodes attached to the same link:
  - Router Discovery: How hosts locate routers that reside on an attached link.
  - Prefix Discovery: How hosts discover the set of address prefixes that define which destinations are on-link for an attached link.
  - Parameter Discovery: How a node learns such link parameters as the link MTU or such Internet parameters as the hop limit value to place in outgoing packets.
  - Address Autoconfiguration: How nodes automatically configure an address for an interface.
  - Address resolution: How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address.

- Next-hop determination: The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent. The next-hop can be a router or the destination itself.

- Neighbor Unreachability Detection: How nodes determine that a neighbor is no longer reachable.

- Duplicate Address Detection: How a node determines that an address it wishes to use is not already in use by another node.

- Redirect: How a router informs a host of a better first-hop node to reach a particular destination.
**Neighbor Discovery Messages**

- **Neighbor Solicitation**: Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Neighbor Solicitations are also used for Duplicate Address Detection.

- **Neighbor Advertisement**: A response to a Neighbor Solicitation message. A node may also send unsolicited Neighbor Advertisements to announce a link-layer address change.

- **Redirect**: Used by routers to inform hosts of a better first hop for a destination.

**Neighbor Discovery (Additional Features)**

- Link-layer address change: A node that knows its link-layer address has changed can multicast a few (unsolicited) Neighbor Advertisement packets to all nodes to quickly update cached link-layer addresses that have become invalid. The Neighbor Unreachability Detection algorithm ensures that all nodes will reliably discover the new address, though the delay may be somewhat longer.

- Inbound load balancing: Nodes with replicated interfaces may want to load balance the reception of incoming packets across multiple network interfaces on the same link. Such nodes have multiple link-layer addresses assigned to the same interface. For example, a single network driver could represent multiple network interface cards as a single logical interface having multiple link-layer addresses.

**Comparison with IPv4**

- The IPv6 Neighbor Discovery protocol corresponds to a combination of the IPv4 protocols ARP, ICMP Router Discovery, and ICMP Redirect. In IPv4 there is no generally agreed upon protocol or mechanism for Neighbor Unreachability Detection.

- Router Discovery is part of the base protocol set.

- Router advertisements and Redirects carry link-layer addresses; no additional packet exchange is needed to resolve the router’s link-layer addresses.

- Router advertisements carry prefixes for a link; there is no need to have a separate mechanism to configure the “netmask”.

- Router advertisements enable Address Autoconfiguration.

- Routers can advertise an MTU for hosts to use on the link.

- Address resolution multicasts are “spread” over 4 billion ($2^{32}$) multicast addresses greatly reducing address resolution related interrupts on nodes other than the target. Moreover, non-IPv6 machines should not be interrupted at all.
### Router Solicitation Message Format

Hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options …</td>
</tr>
</tbody>
</table>

**IP Fields:**
- Source Address: An IP address assigned to the sending interface, or the unspecified address if no address is assigned to the sending interface.
- Destination Address: Typically the all-routers multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender SHOULD include this header.

**ICMP Fields:**
- Type: 133
- Code: 0

**Valid Options:**
- Source link-layer address: The link-layer address of the sender, if known.

### Router Advertisement Message Format

Routers send out Router Advertisement messages periodically, or in response to a Router Solicitation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options …</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Checksum</td>
</tr>
</tbody>
</table>

**IP Fields:**
- Source Address: An address assigned to the interface from which this message is sent.
- Destination Address: Typically the Source Address of an invoking Router Solicitation or the all-nodes multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

**ICMP Fields:**
- Type: 134

**Valid Options:**
- Source link-layer address: The link-layer address of the sender, if known.
- MTU: Should be sent on links that have a variable MTU.
- Prefix Information: These options specify the prefixes that are on-link and/or are used for address autoconfiguration.

### Neighbor Solicitation Message Format

Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target. Neighbor Solicitations are multicast when the node needs to resolve an address and unicast when the node seeks to verify the reachability of a neighbor.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options …</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Target Address (128 bits)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Checksum</td>
</tr>
</tbody>
</table>

**IP Fields:**
- Source Address: Either an address assigned to the interface from which this message is sent or (if Duplicate Address Detection is in progress) the unspecified address.
- Destination Address: Either the solicited-node multicast address corresponding to the target address, or the target address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

**ICMP Fields:**
- Type: 135

**Valid Options:**
- Source link-layer address: The link-layer address of the sender, if known.
- MTU: Should be sent on links that have a variable MTU.
- Prefix Information: These options specify the prefixes that are on-link and/or are used for address autoconfiguration.

### Neighbor Advertisement Message Format

A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>RREP</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options …</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Target Address (128 bits)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Checksum</td>
</tr>
</tbody>
</table>

**IP Fields:**
- Source Address: An address assigned to the interface from which the advertisement is sent.
- Destination Address: For solicited advertisements, the Source Address of an invoking Neighbor Solicitation or, if the solicitation’s Source Address is the unspecified address, the all-nodes multicast address. For unsolicited advertisements typically the all-nodes multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.
Redirect Message Format

Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination. Hosts can be redirected to a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMP Target Address equal to the ICMP Destination Address.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>136</td>
</tr>
<tr>
<td>Code</td>
<td>0</td>
</tr>
<tr>
<td>Target Address</td>
<td>An IP address that is a better first hop to use for the ICMP Destination Address. The Target Address field in the NS message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address is known. Note that on NBMA links, hosts may rely on the presence of the Target Link-Layer Address option in Redirect messages as the means for determining the link-layer addresses of neighbors. In such cases, the option must be included in Redirect messages. Otherwise the target is a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMP Target Address equal to the ICMP Destination Address.</td>
</tr>
<tr>
<td>Source Address</td>
<td>Must be the link-local address assigned to the interface from which this message is sent. (Reserved)</td>
</tr>
<tr>
<td>Destination Address</td>
<td>The Source Address of the packet that triggered the redirect.</td>
</tr>
<tr>
<td>Authentication Header</td>
<td>If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.</td>
</tr>
<tr>
<td>Hop Limit</td>
<td>255</td>
</tr>
</tbody>
</table>

IP Fields:
- Source Address: Must be the link-local address assigned to the interface from which this message is sent.
- Destination Address: The Source Address of the packet that triggered the redirect.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

Neighbor Advertisement Message Format

ICMP Format:
- Type: 136
- Code: 0
- Target Address: An IP address that is a better first hop to use for the ICMP Destination Address. The Target Address field in the NS message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address is known. Note that on NBMA links, hosts may rely on the presence of the Target Link-Layer Address option in Redirect messages as the means for determining the link-layer addresses of neighbors. In such cases, the option must be included in Redirect messages. Otherwise the target is a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMP Target Address equal to the ICMP Destination Address. |
- Source Address: Must be the link-local address assigned to the interface from which this message is sent. (Reserved) |
- Destination Address: The Source Address of the packet that triggered the redirect. |
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header. |
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

Datagram Structures

Red i r ect Message Format

Possible Options:
- Target Link-Layer Address: The link-layer address for the target. It should be included if known. Note that on NBMA links, hosts may rely on the presence of the Target Link-Layer Address option in Redirect messages as the means for determining the link-layer addresses of neighbors. In such cases, the option must be included in Redirect messages. Otherwise the target is a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMP Target Address equal to the ICMP Destination Address. |
- Source Address: Must be the link-local address assigned to the interface from which this message is sent. (Reserved) |
- Destination Address: The Source Address of the packet that triggered the redirect. |
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header. |
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

Summary / Neighbor Discovery

Features
- Router Discovery
- Prefix Discovery Parameter Discovery
- Address Autoconfiguration
- Address resolution
- Next-hop determination
- Neighbor Unreachability Detection
- Duplicate Address Detection
- Redirect

Messages
- Router Solicitation / Router Advertisement
- Neighbor Solicitation / Neighbor Advertisement
- Redirect

Conceptual Model of a Host
Data Structures

Neighbor Cache: A set of entries about individual neighbors to which traffic has been sent recently.
- Neighbor's on-link unicast IP address (key)
- Link-layer address
- IsRouter flag
- Pointer to any queued packets waiting for address resolution to complete
- Reachability state (NUD)
- Number of unanswered probes (NUD)
- Time the next NUD event is scheduled to take place

Destination Cache: A set of entries about destinations to which traffic has been sent recently. Includes both on-link and off-link destinations. IT maps a destination IP address to the IP address of the next-hop neighbor.

Prefix List: A list of the prefixes that define a set of addresses that are on-link.
- Entries are created from information received in Router Advertisements.
- Each entry has an associated invalidation timer value used to expire prefixes when they become invalid.
- A special “infinity” timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement.
**Conceptual Model of a Host**

**Data Structures**

- **Default Router List**: A list of routers to which packets may be sent.
  - Entries point to entries in the Neighbor Cache.
  - The algorithm for selecting a default router favors routers known to be reachable over those whose reachability is suspect.
  - Each entry also has an associated invalidation timer.

---

**Neighbor Cache**

**Neighbor’s Reachability State**

- A key piece of information in the NC is a neighbor’s reachability state, which is one of five possible values:
  - **INCOMPLETE**: Address resolution is in progress and the link-layer address of the neighbor has not yet been determined.
  - **REACHABLE**: The neighbor is known to have been reachable recently (within tens of seconds ago).
  - **STALE**: The neighbor is no longer known to be reachable but until traffic is sent to the neighbor, no attempt should be made to verify its reachability.
  - **DELAY**: The neighbor is no longer known to be reachable, and traffic has recently been sent to the neighbor. Rather than probe the neighbor immediately, however, delay sending probes for a short while in order to give upper layer protocols a chance to provide reachability confirmation.
  - **PROBE**: The neighbor is no longer known to be reachable, and unicast Neighbor Solicitation probes are being sent to verify reachability.

---

**Host Variables**

- In addition the host maintains a number of variables, e.g.:
  - **LinkMTU**: The MTU of the link.
  - **CurHopLimit**: The default hop limit to be used when sending unicast IPv6 packets.
  - **BaseReachableTime**: A base value used for computing the random ReachableTime value.
  - **ReachableTime**: The time a neighbor is considered reachable after receiving a reachability confirmation.
  - **RetransTimer**: The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor.

---

**Conceptual Sending Algorithm**

**Next-Hop Determination**

- Next-hop determination at the sender for a given unicast destination operates as follows:
  - When sending a packet to a destination, a node uses a combination of the Destination Cache, the Prefix List, and the Default Router List to determine the IP address of the appropriate next hop, an operation known as "next-hop determination".
  - Once the IP address of the next hop is known, the Neighbor Cache is consulted for link-layer information about that neighbor.

---

- If the Default Router List is empty, the sender assumes that the destination is on-link.
- For efficiency reasons, next hop determination is not performed on every packet that is sent. Instead, the results of next-hop determination computations are saved in the Destination Cache. When the sending node has a packet to send, it first examines the Destination Cache. If no entry exists for the destination, next-hop determination is invoked to create a Destination Cache entry.
Once the IP address of the next-hop node is known, the sender follows these steps:

1. Examine the Neighbor Cache for link-layer information about that neighbor.
   - If the entry exists, the sender creates one, sets its state to INCOMPLETE, initiate Address Resolution, and then queue the data packet pending completion of address resolution.
   - If no entry exists, the sender transmits the packet.

For multicast-capable interfaces, Address Resolution consists of sending a Neighbor Solicitation message and waiting for a Neighbor Advertisement.

Address resolution is the process through which a node determines the link-layer address of a neighbor given only its IP address.

Address resolution is performed only on addresses that are determined to be on-link and for which the sender does not know the corresponding link-layer address.

Address resolution is never performed on multicast addresses.

When a multicast-capable interface becomes enabled, the node must join the all-nodes multicast address on that interface, as well as the solicited-node multicast address corresponding to each of the IP addresses assigned to the interface.

While waiting for address resolution to complete, the sender must, for each neighbor, retain a small queue of packets waiting for address resolution to complete. Once address resolution completes, the node transmits any queued packets.

While awaiting a response, the sender should retransmit Neighbor Solicitation messages approximately every RetransTimer milliseconds, even in the absence of additional traffic to the neighbor.

If no Neighbor Advertisement is received after MAX_MULTICAST_SOLICIT solicitations, address resolution has failed. The sender must return ICMP destination unreachable indications with code 3 (Address Unreachable) for each packet queued awaiting address resolution.

The recipient should create or update the Neighbor Cache entry for the IP Source Address of the solicitation.

If an entry does not already exist, the node should create a new one and set its reachability state to STALE.

If an entry already exists, and the cached link-layer address differs from the one in the received Source Link-Layer Address option, the cached address should be replaced by the received address and the entry’s reachability state must be set to STALE.

After any updates to the Neighbor Cache, the node sends a Neighbor Advertisement response.
Address Resolution Example

<table>
<thead>
<tr>
<th>Src MAC Addr</th>
<th>Dst MAC Addr</th>
<th>Src IP Addr</th>
<th>Dst IP Addr</th>
<th>ICMP TYPE</th>
<th>Dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:50:56:8a:0:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>Neighbor solicitation</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>0:50:56:8a:0:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>Neighbor advertisement</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>0:50:56:8a:0:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>Echo request</td>
<td>41</td>
<td>4</td>
</tr>
<tr>
<td>0:50:56:8a:0:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>fe80::250:56ff:fe8a:0</td>
<td>Echo reply</td>
<td>41</td>
<td>4</td>
</tr>
</tbody>
</table>

Address Resolution: Sending Solicited Neighbor Advertisements

- A node sends a Neighbor Advertisement in response to a valid Neighbor Solicitation targeting one of the node's assigned addresses.
- The Target Address of the advertisement is copied from the Target Address of the solicitation.
- If the solicitation's IP Destination Address is a multicast address, the Target Link-Layer option must be included in the advertisement.
- If the node is a router, it MUST set the Router flag to one; otherwise it must set the flag to zero.
- If the source of the solicitation is the unspecified address, the node must set the Solicited flag to zero and multicast the advertisement to the all-nodes address. Otherwise, the node must set the Solicited flag to one and unicast the advertisement to the Source Address of the solicitation.

State Machine for the Reachability State

- Address Resolution: Sending Unsolicited Neighbor Advertisements
  - In some cases a node may be able to determine that its link-layer address has changed (e.g., hot-swap of an interface card) and may wish to inform its neighbors of the new link-layer address quickly.
  - In such cases a node may send unsolicited Neighbor Advertisement messages to the all-nodes multicast address.
  - The Target Address field in the unsolicited advertisement is set to an IP address of the interface, and the Target Link-Layer Address option is filled with the new link-layer address. The Solicited flag must be set to zero.
  - Neighboring nodes will immediately change the state of their Neighbor Cache entries for the Target Address to STALE, prompting them to verify the path for reachability.
  - If the Override flag is set to one, neighboring nodes will install the new link-layer address in their caches. Otherwise, they will ignore the new link-layer address, choosing instead to probe the cached address.
Address Resolution: Sending Unsolicited Neighbor Advertisements

- A node that has multiple IP addresses assigned to an interface may multicast a separate Neighbor Advertisement for each address.
- A proxy may multicast Neighbor Advertisements when its link-layer address changes or when it is configured (by system management or other mechanisms) to proxy for an address.
- A node belonging to an anycast address may multicast unsolicited Neighbor Advertisements for the anycast address when the node’s link-layer address changes.
- Because unsolicited Neighbor Advertisements do not reliably update caches in all nodes, they should only be viewed as a performance optimization to quickly update the caches in most neighbors.

Address Resolution: Anycast Neighbor Advertisements

- From the perspective of Neighbor Discovery, anycast addresses are treated just like unicast addresses in most cases.
- Nodes that have an anycast address assigned to an interface treat them exactly the same as if they were unicast addresses with two exceptions.
  - Neighbor Advertisements sent in response to a Neighbor Solicitation should be delayed by a random time between 0 and MAX_ANYPACT_DELAY_TIME to reduce the probability of network congestion.
  - Second, the Override flag in Neighbor Advertisements should be set to 0, so that when multiple advertisements are received, the first received advertisement is used rather than the most recently received advertisement.

Address Resolution: Proxy Neighbor Advertisements

- A router may proxy for one or more other nodes, that is, through Neighbor Advertisements indicate that it is willing to accept packets not explicitly addressed to itself. For example, a router might accept packets on behalf of a mobile node that has moved off-link.
- All solicited proxy Neighbor Advertisement messages must have the Override flag set to zero. This ensures that if the node itself is present on the link its Neighbor Advertisement will take precedence of any advertisement received from a proxy.
- Finally, when sending a proxy advertisement in response to a Neighbor Solicitation, the sender should delay its response by a random time between 0 and MAX_ANYPACT_DELAY_TIME seconds.

Router and Prefix Discovery

- Router Discovery is used to:
  - Locate neighboring routers.
  - Learn prefixes.
  - Learn configuration parameters related to address autoconfiguration.
- Prefix Discovery is the process through which hosts learn the ranges of IP addresses that reside on-link and can be reached directly without going through a router.
- Routers send Router Advertisements that indicate whether the sender is willing to be a default router.
- Router Advertisements also contain Prefix Information options that list the set of prefixes that identify on-link IP addresses.

Neighbor Unreachability Detection

- Communication to or through a neighbor may fail for numerous reasons at any time, including hardware failure, hot-swap of an interface card, etc.
- Thus, a node actively tracks the reachability “state” for the neighbors to which it is sending packets.
- NUD is used for all paths between hosts and neighboring nodes, including host-to-host, host-to-router, and router-to-router.
- When a path to a neighbor appears to be failing, the specific recovery procedure depends on how the neighbor is being used.
  - If the neighbor is the ultimate destination, address resolution should be performed again.
  - If the neighbor is a router, attempting to switch to another router would be appropriate.
- Neighbor Unreachability Detection signals the need for next-hop determination by deleting a Neighbor Cache entry.

Reachability Confirmation

- A neighbor is considered reachable if the node has recently received a confirmation that packets sent recently to the neighbor were received by its IP layer.
- Positive confirmation can be gathered in two ways:
  - Hints from upper layer protocols that indicate a connection is making “forward progress”.
  - Receipt of a Neighbor Advertisement message that is a response to a Neighbor Solicitation message.
- In TCP, for example, receipt of a (new) acknowledgement indicates that previously sent data reached the peer. It is a confirmation that the next-hop neighbor is reachable.
- For off-link destinations, forward progress implies that the first-hop router is reachable.
Redirect Function

Redirect messages are sent by routers to redirect a host to a better first-hop router for a specific destination or to inform hosts that a destination is in fact a neighbor (i.e., on-link).

A router should send a redirect message, subject to rate limiting, whenever it forwards a packet that is not explicitly addressed to itself in which:
- The Source Address field of the packet identifies a neighbor, and
- the router determines that a better first-hop node resides on the same link as the sending node for the Destination Address of the packet being forwarded, and
- the Destination Address of the packet is not a multicast address.

The transmitted redirect packet:
- In the Target Address field: the address to which subsequent packets for the destination should be sent.
- In the Destination Address field: the destination address of the invoking IP packet.

In the options:
- Target Link-Layer Address option: link-layer address of the target, if known.
- Redirected Header: as much of the forwarded packet as can fit without the redirect packet exceeding 1280 octets in size.

A router must limit the rate at which Redirect messages are sent.
A router must not update its routing tables upon receipt of a Redirect.

Off-link destination

Other Networks

IPv6 Datagram
IPv6 Src Addr = A
IPv6 Dst Addr = B

Redirect Message
IPv6 Src Addr = R1
IPv6 Dst Addr = A
Redir Target Addr = B
Redir Dst Addr = B

R1 forwards the datagram to B anyway

Path MTU Discovery

The Path MTU is the minimum link MTU of all the links in a path between a source node and a destination node.

Path MTU Discovery is the process by which a node learns the PMTU of a path.

The basic idea is that a source node initially assumes that the PMTU of a path is the (known) MTU of the first hop in the path.

If any of the packets sent on that path are too large to be forwarded by some node along the path, that node will discard them and return ICMPv6 Packet Too Big messages.

Upon receipt of such a message, the source node reduces its assumed PMTU for the path based on the MTU of the constricting hop as reported in the Packet Too Big message.

The Path MTU Discovery process ends when the node's estimate of the PMTU is less than or equal to the actual PMTU.
The PMTU of a path may change over time, due to changes in the routing topology.
- Reductions of the PMTU are detected by Packet Too Big messages.
- To detect increases in a path's PMTU, a node periodically increases its assumed PMTU.

In the case of a multicast destination, the PMTU is the minimum PMTU value across the set of paths in use.

The TCP layer must track the PMTU for the path(s) in use by a connection; it should not send segments that would result in packets larger than the PMTU.

---

IPv6 defines both a stateful and stateless address autoconfiguration mechanism.

Stateless autoconfiguration requires:
- no manual configuration of hosts
- minimal (if any) configuration of routers
- no additional servers

The stateless mechanism allows a host to generate its own addresses using a combination of:
- Locally available information (interface identifier)
- Information advertised by routers (link prefixes)

In the absence of routers, a host can only generate link-local addresses. Link-local addresses are sufficient for allowing communication among nodes attached to the same link.

---

In the stateful autoconfiguration model hosts obtain from a server:
- Interface addresses
- Configuration information and parameters.
- Both.

Stateful servers maintain a database that keeps track of which addresses have been assigned to which hosts.

Stateless and stateful autoconfiguration complement each other.

The stateless approach is used when a site is not particularly concerned with the exact addresses hosts use, so long as they are unique and properly routable.

The stateful approach is used when a site requires tighter control over exact address assignments.

Both stateful and stateless address autoconfiguration may be used simultaneously.

---

IPv6 addresses are leased to an interface for a fixed (possibly infinite) length of time.

Each address has an associated lifetime that indicates how long the address is bound to an interface.

When a lifetime expires, the address may be reassigned to another interface elsewhere.

To handle the expiration of address bindings gracefully, an address goes through two distinct phases while assigned to an interface:
- Initially, it is "preferred" (its use in arbitrary communication is unrestricted).
- Later, an address becomes "deprecated" in anticipation that its current interface binding will become invalid.

A deprecated address should be used only by applications that have been using it and would have difficulty switching to another address without a service disruption.

---

Invalid address: an Address that is not assigned to any interface.

Tentative address: An address whose uniqueness on a link is being verified, prior to its assignment to an interface.

Preferred address: An address assigned to an interface whose use is encouraged, but not forbidden. A preferred address should no longer be used as a source address in new communications.

Valid address: A preferred or deprecated address.
Stateless Address Autoconfiguration

Example / Zebra RA Config

interface eth0
no ipv6 nd suppress-ra
ipv6 nd ra-interval 60
ipv6 nd prefix-advertisement 3ffe:38e1:100:4001::/64 2592000 604800 onlink autoconfig
!
interface eth1
no ipv6 nd suppress-ra
ipv6 nd ra-interval 60
ipv6 nd prefix-advertisement 3ffe:38e1:100:7001::/64 2592000 604800 onlink autoconfig
!
interface eth2
no ipv6 nd suppress-ra
ipv6 nd ra-interval 60
ipv6 nd prefix-advertisement 3ffe:38e1:100:2001::/64 2592000 604800 onlink autoconfig

Duplicate Address Detection

Example: Successful Assignment

Duplicate Address Detection

Example: Duplicated Assignment

Stateless Address Autoconfiguration

Site Renumbering

- Address leasing facilitates site renumbering by providing a mechanism to time-out addresses assigned to interfaces in hosts.

- Dividing valid addresses into preferred and deprecated categories provides a way of indicating to upper layers that a valid address may become invalid shortly and that future communication using the address will fail.

- To avoid this scenario, higher layers should use a preferred address to increase the likelihood that an address will remain valid for the duration of the communication.

- The deprecation period should be long enough that most, if not all, communications are using the new address at the time an address becomes invalid.

Duplicate Address Detection

Example: Successful Assignment

Duplicate Address Detection

Example: Duplicated Assignment

Stateless Address Autoconfiguration

Example: Successful Assignment

Duplicate Address Detection

Example: Duplicated Assignment

Stateless Address Autoconfiguration

Example: Successful Assignment

Duplicate Address Detection

Example: Duplicated Assignment
Summary / Address Autoconfiguration

- No manual config of hosts
- Hosts generate their own addresses (using RA)
- Stateful and Stateless Autoconfig complement each other
- Addresses are leased for a fixed length of time.
- Duplicate Address Detection is performed
- Address States: Invalid / Tentative / Preferred / Deprecated
- Allows Renumbering

Multicast Listener Discovery

- Enables each IPv6 router to discover the presence of multicast listeners on its attached links.
- Discovers specifically which multicast addresses are of interest to those neighboring nodes.
- This information is then provided to whichever multicast routing protocol is being used.
- There are three types of MLD Messages:
  - Multicast Listener Query (ICMPv6 Type 130)
    - General
    - Multicast-Address-Specific Query
  - Multicast Listener Report (ICMPv6 Type 131)
    - Multicast Listener Done (ICMPv6 Type 132)

Multicast Listener Discovery Message Format

- MLD is a sub-protocol of ICMPv6, messages have the following format:

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Maximum Response Delay</th>
<th>Checksum</th>
<th>Reserved</th>
<th>Multicast Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>8</td>
<td>16 bits</td>
<td>16 bits</td>
<td>16 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

IP Fields:
- Source Address: Must be a link-local address assigned to the interface from which this message is sent.
- Hop Limit: 1
- Router Alert in a Hop-by-Hop Options header.

ICMPv6 Fields:
- Maximum Response Delay: Meaningful only in Query messages. Specifies the maximum allowed delay before sending a responding Report (milliseconds).
- Multicast Address: Set to a specific IPv6 multicast address in a Multicast-Address-Specific Query, Report or Done Message.

Multicast Listener Discovery Message Destinations

<table>
<thead>
<tr>
<th>Message Type</th>
<th>IPv6 Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Query</td>
<td>Link-scope all-nodes (FF02::1)</td>
</tr>
<tr>
<td>Multicast-Address-Specific Query</td>
<td>The multicast address being queried</td>
</tr>
<tr>
<td>Report</td>
<td>The multicast address being reported</td>
</tr>
<tr>
<td>Done</td>
<td>Link-scope all-routers (FF02::2)</td>
</tr>
</tbody>
</table>

Multicast Listener Discovery Node Behavior

- A node may be in one of three possible states with respect to any single IPv6 multicast address on any single interface:
  - Non-Listener: when the node is not listening to the address on the interface.
  - Delaying Listener: when the node is listening to the address on the interface and has a report delay timer running for that address.
  - Idle Listener: when the node is listening to the address on the interface and does not have a report delay timer running for that address.
There are seven possible actions that may be taken in response to the events:

- **Send report**: for the address on the interface. The Report message is sent to the address being reported.
- **Send done**: for the address on the interface. If the flag saying we were the last node to report is cleared, this action may be skipped. The Done message is sent to the link-scope all-routers address (FF02::2).
- **Set flag**: that we were the last node to send a report for this address.
- **Clear flag**: since we were not the last node to send a report for this address.
- **Start timer**: for the address on the interface, using a delay value chosen uniformly from the interval \([0, \text{Maximum Response Delay}]\), where Maximum Response Delay is specified in the Query. If this is an unsolicited Report, the timer is set to a delay value chosen uniformly from the interval \([0, \text{[Unsolicited Report Interval]}]\).
- **Reset timer**: for the address on the interface to a new value, using a delay value chosen uniformly from the interval \([0, \text{[Unsolicited Report Interval]}]\), as described in “start timer”.
- **Stop timer**: for the address on the interface.

---

There are three actions that may be taken in response to the events:

- **Start general query timer**: for the attached link to \([\text{Query Interval}]\).
- **Start other querier present timer**: for the attached link to \([\text{Other Querier Present Interval}]\).
- **Send general query**: on the attached link. The General Query is sent to the link-scope all-nodes address (FF02::1), and has a Maximum Response Delay of \([\text{Query Response Interval}]\).

---

To keep track of which multicast addresses have listeners, a router may be in one of three possible states with respect to any single IPv6 multicast address on any single attached link:

- **No Listeners Present**: when there are no nodes on the link that have sent a Report for this multicast address. This is the initial state for all multicast addresses on the router.
- **Listeners Present**: when there is a node on the link that has sent a Report for this multicast address.
- **Checking Listeners**: when the router has received a Done message but has not yet heard a Report for the identified address.
Multicast Listener Discovery
Router Behavior

There are five significant events that can cause router state transitions:

- **Report received**: occurs when the router receives a Report for the address from the link.
- **Done received**: occurs when the router receives a Done message for the address from the link.
- **Multicast-address-specific query received**: occurs when a router receives a Multicast-Address-Specific Query for the address from the link.
- **Timer expired**: occurs when the timer set for a multicast address expires.

There are seven possible actions that may be taken in response to the events:

- **Start timer** for the address on the link.
- **Start timer** for the address on the link - this alternate action sets the timer to the minimum of its current value and either [Last Listener Query Interval] * [Last Listener Query Count] if this router is a Querier, or the Maximum Response Delay in the Query message * [Last Listener Query Count] if this router is a non-Querier.
- **Start retransmit timer** for the address on the link.
- **Clear retransmit timer** for the address on the link.
- **Send multicast-address-specific query** for the address on the link.
- **Notify routing +** internally notify the multicast routing protocol that there are listeners to this address on this link.
- **Notify routing -** internally notify the multicast routing protocol that there are no longer any listeners to this address on this link.

Multicast Listener Discovery
Router State Transition Diagram (Group State)

The following state diagram apply per group per link in Querier state.

- **No Listeners Present**:
  - Report received (send query, clear timer)
  - Done received from (start timer)
  - Timer expired (notify routing+)

- **Listeners Present**:
  - Report received (start timer*)
  - Done received from (start timer*, start retransmit timer, send m-a-s query)
  - Timer expired (clear retransmit timer)
  - Multicast-address-specific query received (send m-a-s query, start retransmit timer)

References:
Event | Action
--- | ---
Report received | start timer
Done received | start timer*, start retransmit timer, send m-a-s query
Timer expired | notify routing+
Multicast-address-specific query received | send m-a-s query, start retransmit timer

The following state diagram apply per group per link in Non-Querier state.

- **No Listeners Present**:
  - Report received (notify routing-)
  - Done received from (start timer*, start retransmit timer)
  - Timer expired (notify routing-, clear retransmit timer)

- **Listeners Present**:
  - Report received (start timer*)
  - Done received from (start timer*, start retransmit timer, send m-a-s query)
  - Timer expired (clear retransmit timer)
  - Multicast-address-specific query received (send m-a-s query, start retransmit timer)

References:
Event | Action
--- | ---
Report received | start timer*
Done received | start timer*, start retransmit timer, send m-a-s query
Timer expired | notify routing-
Multicast-address-specific query received | send m-a-s query, start retransmit timer

Summary / MLD

- Discover Multicast Listeners
- Discover Multicast Groups -> Inform to the MRP
- Messages: Query / Report / Done
- Node States: Non Listener / Delaying Listener / Idle Listener
- Router States: Querier / Non-Querier

IPv6 over Ethernet

- IPv6 packets are transmitted in standard Ethernet frames.
  - The Ethernet header contains:
    - Destination and Source Ethernet addresses.
    - Ethernet type code (86DD hexadecimal).
  - The data field contains:
    - The IPv6 header.
    - The payload.
    - Padding octets to meet the minimum frame size for the Ethernet link (if needed).
  - The Maximum Transmission Unit (MTU) for Ethernet is 1500 octets.
  - This size may be reduced by:
    - A Router Advertisement.
    - Manual configuration of each node.
**IPv6 over Ethernet Frame Format**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Source</th>
<th>Type</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>46-1500</td>
<td>4</td>
</tr>
</tbody>
</table>

- **Preamble:** 1010...1011
- **Destination:** Destination Node Address
- **Source:** Source Node Address
- **Type:** Higher Layer protocol Type
- **Data:** Higher Layer Information
- **FCS:** Frame Check Sequence (CRC-32)

**IPv6 Header and Payload**

- **IPv6:** 64-1518 octets
- **Ethernet:** 128 bits

**Example:**

- EUI-48 Ethernet Address: 00:50:56:d9:88:3f
- EUI-64 Ethernet Address: 00:50:56:ff:fe:d9:88:3f
- Interface Identifier: 02:50:56:ff:fe:d9:88:3f
- Link-local Address: fe80::250:56ff:fed9:883f

**IPv6 over Ethernet / Stateless Autoconfiguration**

- The Interface Identifier for an Ethernet interface is based on the EUI-64 identifier derived from the interface's built-in 48-bit IEEE 802 address.
- The Interface Identifier is then formed from the EUI-64 by complementing the "Universal/Local" (U/L) bit, which is the next-to-lowest order bit of the first octet of the EUI-64.
- For example, the Interface Identifier for an Ethernet interface whose built-in address is, in hexadecimal,

  34-56-78-9A-BC-DE

  would be

  36-56-78-FF-FE-9A-BC-DE

- An IPv6 address prefix used for stateless autoconfiguration of an Ethernet interface must have a length of 64 bits.

**IPv6 over Ethernet / Link-local Address**

- The IPv6 link-local address for an Ethernet interface is formed by appending the Interface Identifier, to the prefix FE80::/64.

  - **Bits:**
    - 111111101011 0 10 bits Interface Identifier from Ethernet Address
    - 54 bits 64 bits

  - **Example:**
    - EUI-48 Ethernet Address: 00:50:56:d9:88:3f
    - EUI-64 Ethernet Address: 00:50:56:ff:fe:d9:88:3f
    - Interface Identifier: 02:50:56:ff:fe:d9:88:3f
    - Link-local Address: fe80::250:56ff:fed9:883f

**IPv6 over Ethernet / Unicast address mapping**

- The Neighbor Discovery Source/Target Link-layer Address option has the following form when the link layer is Ethernet.

  - **Type:** 8 bits
  - **Length:** 8 bits
  - **Ethernet Address:** 48 bits

  - **Example:**
    - Type: 1 for Source Link-layer address, 2 for Target Link-layer address
    - Ethernet Address: The 48 bit Ethernet IEEE 802 address, in canonical bit order

**IPv6 over Ethernet / Multicast address mapping**

- An IPv6 packet with a multicast destination address DST, consisting of the sixteen octets DST[1] through DST[16], is transmitted to the Ethernet multicast address whose first two octets are the value 3333 hexadecimal and whose last four octets are the last four octets of DST.

  - **Example:**
    - IPv6 Solicited-Node Multicast Address: f02:1::1:ff09:883f
    - Ethernet Link-Layer Multicast Address: 33:33:ff:09:88:3f
    - IPv6 All nodes Multicast Address: f02:1::1
    - Ethernet Link-Layer Multicast Address: 33:33:00:00:00:1

**IPv6 over PPP**

- The Point-to-Point Protocol (PPP) has three main components:
  - A method for encapsulating datagrams over serial links.
  - A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
  - A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.

- The NCP for establishing and configuring the IPv6 over PPP is called the IPv6 Control Protocol (IPv6CP).
Exactly one IPv6 packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 0057 (Internet Protocol Version 6).

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Protocol</th>
<th>Information (IPv6 or IPv6CP)</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Protocol:
- 0057 hex: IPv6
- 0057 hex: IPv6 Control Protocol

IPV6CP Configuration Options

IPV6CP Configuration Options allow negotiation of desirable IPv6 parameters.

Current values are assigned as follows:

1. Interface-Identifier.
   This Configuration Option provides a way to negotiate a unique 64-bit interface identifier to be used for the address autoconfiguration at the local end of the link.

2. IPv6-Compression-Protocol.
   This Configuration Option provides a way to negotiate the use of a specific IPv6 packet compression protocol. The IPv6-Compression-Protocol Configuration Option is used to indicate the ability to receive compressed packets. Each end of the link must separately request this option if bi-directional compression is desired. By default, compression is not enabled.

Link-local Address

The interface identifier may be selected using one of the following methods:

- If an IEEE global identifier is available anywhere on the node, then that address should be used.
- If an IEEE global identifier is not available, then a different source of uniqueness, such as a machine serial number, should be used.
- If a good source of uniqueness cannot be found, a random number should be generated.

The resulting link-local address is:

```
``