Thomas Narten, IBM
Erik Nordmark, Sun Microsystems
W A Simpson, Daydreamer

IPv6 Neighbor Discovery

<draft-ietf-ipngwg-discovery-00.txt>

Status of this Memo

Abstract

This document specifies the Neighbor Discovery protocol for the IP Version 6 protocol. IPv6 nodes on a single link use Neighbor Discovery to discover each other's presence and to determine each other's link-layer addresses.

Contents

1. INTRODUCTION

This specification defines the Neighbor Discovery (ND) protocol for the IP Version 6 protocol. Nodes (hosts and routers) use Neighbor Discovery to determine the link-layer address for neighbors known to reside on attached links and to quickly learn new link-layer addresses should cached values become invalid. Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses. Sending hosts also detect when routers fail and actively search for functioning alternates.

This document is a new revision of the protocol specified in the two

documents:
<draft-simpson-ipv6-discov-formats-02.txt>, and
<draft-simpson-ipv6-discov-process-02.txt>

The authors would like to acknowledge the contributions of (in alphabetical order) Ran Atkinson, Scott Bradner, Stephen Deering, Robert Hinden, Allison Mankin, Dan McDonald, and Sue Thomson.

2. TERMINOLOGY

node - a device that implements IPv6.

router - a node that forwards IPv6 packets not explicitly addressed to itself.

host - any node that is not a router.

upper layer - a protocol layer immediately above IPv6. Examples are transport protocols such as TCP and UDP, control protocols such as ICMP, routing protocols such as OSPF, and internet or lower-layer protocols being "tunneled" over (i.e., encapsulated in) IPv6 such as IPX, AppleTalk, or IPv6 itself.

link - a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.

interface - a node's attachment to a link.

neighbors - nodes attached to the same link.

on-link - a destination node that is a neighbor to the sender. A host considers a destination to be on-link if:
- it is covered by one of the link's prefixes, or
- a neighboring router specifies the destination as the target of a Redirect message, or
- a Neighbor Advertisement message is received from the address, or
- a Router Advertisement message is received from the address.

off-link - the opposite of "on-link"; a destination node that is

not a neighbor to the sender.

address - an IPv6-layer identifier for an interface or a set of interfaces.

designated address
- one of an interface's assigned addresses on a router. It is used as the source address in all Neighbor Discovery messages sent from the interface, and neighboring nodes use the address to uniquely identify a router's interface. The designated address should only change infrequently.

anycast address
- an identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance). See [ADDR-ARCH].

link-layer address
- a link-layer identifier for an interface. Examples are IEEE 802 addresses for Ethernet links, E.164 addresses for ISDN links.

reachability
- whether or not two IPv6 nodes can communicate with each other. For routers reachability means that packets sent by a node's IPv6 layer are delivered to the router's IPv6 layer, and the router is indeed forwarding the packets (i.e. it has not been converted to a host). For hosts, reachability means that packets sent by a node's IPv6 layer are delivered to the neighbor host's IPv6 layer. Note that reachability only applies to the "forward" path from one neighbor to another.

packet - an IPv6 header plus payload.

link MTU - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed in one piece over a link.

target - a node about which address resolution information is sought, or a node which is the new first-hop when being redirected.

proxy - a router that responds to Neighbor Discovery query messages on behalf of another node. For instance, a router that is bound to an anycast address will respond on behalf of the anycast address, and potentially a router acting on behalf of a mobile node, that has moved off-link, act as a proxy for the mobile node.

Different link layers have different properties. The ones of concern to Neighbor Discovery are:

point-to-point - a link that has exactly two interfaces.

multicast - a link that supports some mechanism at the link layer for sending packets to all (i.e. broadcast) or a subset of all neighbors. Multicast/broadcast can be provided by a multitude of link layer mechanisms such as the physical link layer itself (for example, Ethernet), replicated unicast packets sent by the link layer software, or multicast servers (such as in ATM).

non-broadcast multiple access (NBMA) - a link with more than two neighbors that does not support any form of multicast or broadcast (e.g., Frame Relay).

shared media - a link that allows direct communication among a number of nodes, but the nodes do not all share the same IP address prefix, and may not know which nodes are on-link. Examples are large (switched) public data networks such as SMDS and B-ISDN. Also known as "large clouds". See [SH-MEDIA].

variable MTU - a link that does not have a well-defined MTU. For example Token Ring (IEEE 802.5).

asymmetric connectivity - a link where non-reflexive and/or non-transitive connectivity is part of normal operation. (Non-reflexive connectivity is when A can hear B but B can't hear A. Non-transitive connectivity is when A can hear B, and B can hear C, but A can't hear C.) Many radio links exhibit these properties.

Neighbor Discovery makes use of a number of different addresses defined in [ADDR-ARCH], including:

all-nodes multicast address
- the link scope address to reach all nodes. FF02::1

all-routers multicast address
- the link scope address to reach all routers. FF02::2

all-hosts multicast address
- the link scope address to reach all hosts. FF02::3

solicited-node multicast address
- a multicast address that is computed as a function of the solicited target's address. The solicited-node multicast address is formed by taking the bit-wise exclusive-or of all octets in the IP address and producing a hash value between 0 and 255. The hash value is appended to the 15-octet prefix FF02::07, resulting in a multicast address in the range FF02::0700 to FF02::07FF. For example: the solicited node multicast address corresponding to the IP address 4037::01:800:200E:8C6C is FF02::07B0.

unspecified address
- the address 0:0:0:0:0:0:0:0 . It indicates the absence of an address. One example of its use is in the Source Address field of any IPv6 packets sent by an initializing host before it has learned its own address.

3. PROTOCOL OVERVIEW

This protocol solves a set of problems related to the interaction between nodes attached to the same shared link. It defines mechanisms for solving each of the following problems:

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. (Nodes use prefixes to distinguish destinations that reside on-link from those only reachable through a router.)

Address Autoconfiguration: How nodes automatically configure an address for an interface.

Address Resolution: How nodes determine the link-layer address of a neighboring node given only the node'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.

Neighbor Unreachability Detection: How nodes determine that a neighbor is no longer reachable. For neighbors used as routers, alternate default routers can be tried. For both routers and hosts, Address Resolution can be performed again.

Duplicate Address Detection: How a node detects if another node has been configured with the same address.

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

Neighbor Discovery defines five different ICMPv6 packet types: A pair of Router Solicitation and Router Advertisement messages, a pair of Neighbor Solicitation and Neighbor Advertisements messages, and a Redirect message. The messages serve the following purpose:

Router Solicitation: When an interface becomes enabled, hosts may send out Router Solicitations that force routers to generate Router Advertisements immediately rather than at their next scheduled time.

Router Advertisement: Routers advertise their presence together with various link parameters either periodically, or in response to an explicit Router Solicitation message.

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 can also be 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.

Each router periodically multicasts a Router Advertisement packet

announcing its availability. A host receives Router Advertisements from all routers, building a prioritized list of routers that can be used as defaults. Routers generate Router Advertisements frequently enough that hosts will learn of their presence within a few minutes, but not frequently enough to rely on an absence of advertisements to detect router failure; a separate Neighbor Unreachability Detection algorithm handles this condition.

Router Advertisements contain a list of on-link prefixes. Hosts use advertised prefixes to build and maintain a list of current on-link prefixes, which are used in deciding when a packet's destination is on-link or behind a router. Note that a destination can be on-link even though it is not covered by any advertised prefixes. In such cases a router will send a Redirect informing the sender that the destination is a neighbor.

Prefix information contained in Router Advertisement messages includes additional information used by Address Autoconfiguration. This allows the routers to specify if hosts should use stateful (DHCPv6) or autonomous (stateless) address configuration. The Router Advertisement messages also specify lifetimes for addresses that are configured using autonomous address configuration. The exact semantics and usage of the address configuration-related information is specified in [ADDRCONF].

Address Resolution is accomplished by multicasting a Neighbor Solicitation query asking the target node to return its link-layer address. Neighbor Solicitation messages are multicast to the solicited-node multicast address corresponding to the target address. The target returns its link-layer address in a unicast Neighbor Advertisement message. A single request-response pair of packets is sufficient for both the initiator and the target to resolve each other's link-layer addresses; the initiator includes its link-layer address in the Neighbor Solicitation query.

Neighbor Solicitation messages can also be used to determine if another node has been configured to use a particular address. The use of Neighbor Solicitation messages for Duplicate Address Detection is specified in [ADDRCONF].

Neighbor Unreachability Detection requires positive confirmation that packets sent to a neighbor are actually reaching that neighbor and being processed properly by its IPv6 layer. Neighbor Unreachability Detection uses confirmation from two sources. When possible, upper-layer protocols provide a positive confirmation when a connection is making "forward progress", that is, previously sent data is known to have been delivered correctly (e.g., new acknowledgments were received recently). When positive confirmation is not forthcoming through such "hints", a node sends explicit unicast Neighbor Solicitation messages that solicit

Neighbor Advertisement a reachability confirmation from the next hop. To reduce unnecessary network traffic, probe messages are only sent to neighbors to which the node is sending packets.

In addition to addressing the above general problems, Neighbor Discovery also handles the following situations:

Link-layer address change - A node that knows its link-layer address has changed can multicast a few Neighbor Advertisement packets to all nodes to quickly (but unreliably) 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 will be somewhat longer.

Proxy advertisements - A router willing to accept packets on behalf of a node that is unable to respond to Neighbor Solicitations can issue proxy Neighbor Advertisements. Proxy advertising is currently only defined for use with anycast addresses, but could potentially also be used to handle mobile nodes that have moved off-link. However, it is not intended as a general mechanism to handle nodes that e.g. do not implement this protocol. Nodes are aware when a proxy is being used, and the protocol allows multiple proxies to serve the same target. When multiple advertisements are received, rules specify precedence and how to break ties.

Anycast addresses - Anycast addresses identify one of a set of routers providing an equivalent service, and multiple routers on the same net may be configured to recognize the same Anycast address. A Neighbor Advertisement for an Anycast address will, just like other proxy advertisement, contain a source IP address that differs from the target address, and nodes will process them in the same manner they process other proxy advertisements.

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. Routers may omit the source link-layer address from Router Advertisement packets, forcing neighbors to use Neighbor Solicitation messages to learn the link-layer addresses. Returned Neighbor Advertisement messages can then contain different link-layer addresses dependent on who issued the query.

3.1. Comparison with IPv4

The IPv6 Neighbor Discovery protocol corresponds to a combination of the IPv4 protocols ARP [ARP], ICMP Router Discovery [RDISC], and ICMP Redirect [ICMPv4]. In IPv4 there is no generally agreed upon protocol mechanism for Neighbor Unreachability Detection, all though the Hosts Requirements [HR-CL] does specify some possible algorithms for Dead Gateway Detection (which is a subset of Neighbor Unreachability Detection).

The Neighbor Discovery protocol provides a multitude of improvements over the IPv4 set of protocols:

Router Discovery is part of the base; no need for hosts to "snoop" the routing protocols.

Router advertisements carry link-layer address; no additional packet exchange is needed to resolve the router's link-layer address.

Router advertisements carry prefixes for a link; no need to have some other mechanism to configure the "netmask".

Router advertisements contain hooks for Address Autoconfiguration.

By default, hosts learn all on-link prefixes from Router Advertisements. However, routers may be configured to omit some or all prefixes from Router Advertisements. In such cases hosts will assume that destinations are off-link and send traffic to routers by default. A router can then issue redirects for on-link destinations as appropriate. This mechanism may be useful with shared media links where hosts might not know all the on-link prefixes.

Routers can advertise an MTU for hosts to use on the link; better handling of links without a well-defined MTU.

Address Resolution uses multicast "spread" over 256 multicast addresses; greatly reduced Address Resolution related interrupts for nodes other than the target and generates no interrupts on non-IPv6 nodes.

Redirects contain the link-layer address of the new first hop; separate Address Resolution is not needed upon receiving a redirect.

Nodes assume the new next-hop target address in a Redirect is on-link making it possible to redirect to targets that do not share a

common address prefix with the sender. This is an implementation of the XRedirect idea in [SH-MEDIA] and it simplifies some aspects of neighbor interaction on shared media.

Neighbor Unreachability Detection is part of the base. This significantly improves the robustness of packet delivery in the presence of failing routers and nodes that change their link-layer addresses. For instance, it enables mobile nodes to move off-link without loosing any connectivity due to stale ARP caches.

Placing address resolution at the ICMP layer makes the protocol more media-independent than ARP and makes it possible to use standard IPv6 authentication and security mechanisms as appropriate [IPv6-AUTH, IPv6-ESP].

3.2. Supported Link Types

Neighbor Discovery supports links with different properties. In the presense of certain properties only a subset of the ND protocol is available:

point-to-point - Neighbor Discovery handles such links just like multicast links. (Multicast can be trivially provided on point to point links.)

multicast - All aspects of Neighbor Discovery are available.

non-broadcast multiple access (NBMA)
- The only Neighbor Discovery mechanism available on these links is Redirect.

  If the hosts support manual configuration of a list of default routers the hosts can dynamically acquire the link-layer addresses for their neighbors from Redirect messages.

shared media - The Redirect message is modeled after the XRedirect message in [SH-MEDIA] in order to simplify use of the protocol on shared media links.

  This specification does not address shared media issues that only relate to routers, such as:

  - How routers exchange reachability information on a shared media link.

- How a router determines the link-layer address of a hosts, which it needs to send redirect messages to the host.

- How a router determines that it is the first hop router for a received packet.

The protocol is extensible (using extensions) so that other solutions might be possible in the future.

variable MTU - Neighbor Discovery allows the routers to specify a MTU for the link. This allows all nodes to use the same MTU. Note: It is not possible to have each node use a different MTU (or Maximum Receive Unit) due to multicast.

asymmetric connectivity

- Neighbor Discovery detects the absence of symmetric connectivity; a node avoids using a neighbor with which it does not have symmetric connectivity.

  The protocol can presumably be extended in the future to find viable paths in environments that lack reflexive and transitive connectivity.

4. CONCEPTUAL MODEL OF A HOST

This section describes a conceptual model of one possible data structure organization that hosts (and to some extent routers) will maintain in interacting with neighboring nodes. The described organization is provided to facilitate the explanation of how the Neighbor Discovery protocol should behave. This document does not mandate that implementations adhere to this model as long as their behavior is consistent with the protocol specification.

This model is only concerned with the aspects of host behavior directly related to Neighbor Discovery. In particular, it does not concern itself with issues like source address selection and selecting the outgoing interface on a multihomed host.

4.1. Conceptual Data Structures

Hosts will need to maintain the following pieces of information about an interface:

Neighbor Cache - A set of entries about individual neighbors to which traffic has been sent recently. Entries are keyed on the neighbor's IP address and contain such information as its link-layer address, a flag indicating whether the neighbor is a router or a host (called "is_router" in this document), a pointer to any queued packets waiting for Address Resolution to complete, etc.

A Neighbor Cache entry also contains information used by the Neighbor Unreachability Detection algorithm. This includes the reachability state, the number of unanswered probes, and the earliest time the next probe can be sent.

Next-Hop Cache - A set of entries for each destination to which traffic has been sent recently. The Next-Hop Cache includes both on-link and off-link destinations and provides a level of indirection into the Neighbor Cache; the Next-Hop Cache maps a destination IP address to the IP address of the next-hop neighbor. Implementations may find it convenient to store additional information not directly related to Neighbor Discovery in next-hop entries, such as the Path MTU (PMTU) and round trip timers maintained by transport protocols.

Prefix List - A list of the prefixes that define a set of IP addresses that are on-link. Prefix list entries are created from information received in Router Advertisements. Each entry has an associated invalidation timer value (extracted from the advertisement) used to delete prefixes that routers stop advertising.

Default Router List
- A list of routers, prioritized by preference, to which packets may be sent. Router list entries will point to entries in the Neighbor Cache so that when a router is being selected, routers known to be reachable can be favored over those whose reachability is suspect. Each entry also has an associated timer value (extracted from Router Advertisements) used to delete entries that are not longer advertised.

The Neighbor Cache contains information maintained by Neighbor Unreachability Detection algorithm. A key piece of information is a

neighbor's reachability status which is described by one of three values:

REACHABLE Roughly speaking, the neighbor is known to have been reachable recently (within tens of seconds ago).

PROBE The neighbor is probably reachable, but the last explicit reachability confirmation was received long enough ago that verification is now actively sought.

TRY_ALTERNATES
Several attempts at verifying reachability have failed, and additional attempts at restoring communication are warranted. If the neighbor is being used as a router, for example, an alternate router might be tried. Alternatively, if the neighbor is the destination, Address Resolution can be performed again to detect a potentially changed link-layer address.

4.2. Conceptual Sending Algorithm

When sending a packet, a node uses a combination of the Next-Hop 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-level information about that neighbor.

Next-hop determination operates as follows. The sender examines the Prefix List to determine whether the packet's destination is on- or off-link. If the destination is on-link the sender knows that the next-hop address is the same as the packet's destination address. If the destination is off-link, the sender selects a router from the Default Router List (following the rules described in Section 5.3.3). If there are no routers on the Default Router List, 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 Next-Hop Cache. When the sending node has a packet to send, it first examines the Next-Hop Cache. If no entry exists for the destination, next-hop determination is invoked to create a Next-Hop Cache entry.

Once the IP address of the next-hop node is known, the sender examines the Neighbor Cache for link-level information about that neighbor. If no entry exists, the node creates one, sends a Neighbor Solicitation

query, and queues the packet pending completion of Address Resolution. When a Neighbor Advertisement response is received, the link-layer addresses is entered in the Neighbor Cache entry and the queued packet is transmitted. This Address Resolution mechanism is described in section 6.2.

Each time a Neighbor Cache entry is accessed to transmit a packet, the sender checks Neighbor Unreachability Detection related information according to the Neighbor Unreachability Detection algorithm (section 6.3). This check might result in the sender transmitting a Neighbor Solicitation to verify that the neighbor is still reachable.

Next-hop determination is done the first time traffic is sent to a destination. As long as subsequent communication to that destination proceeds successfully, the Next-Hop Cache entry continues to be used. If at some point communication ceases to proceed, as determined by the Neighbor Unreachability Detection algorithm, next-hop determination may need to be performed again. For example, traffic through a failed router should be switched to a working router. Likewise, it may be possible to reroute traffic destined for a mobile node to a "mobility agent".

Note that when a node redoes next-hop determination there is no need to discard the complete Next-Hop Cache entry. In fact, it is often beneficial to retain information, such as cached PMTU and round trip timer values, that are kept in the Next-Hop Cache entry.

4.3. Garbage Collection and Timeout Requirements

The conceptual data structures described above use different mechanisms for discarding potentially stale, as well as unused, information.

>From the perspective of correctness, there is no need to periodically purge Next-Hop and Neighbor Cache entries. Although stale information can potentially remain in the cache indefinitely, the Neighbor Unreachability Detection algorithm described in this document insures that stale information is purged quickly if it is actually being used.

To limit the storage needed for the Next-Hop and Neighbor Caches, a node may need to garbage-collect old entries. However, care must be taken to insure that sufficient space is always present to hold the working set of active entries. A small cache may result in an excessive amount of Neighbor Discovery messages as discarded entries are rebuilt. Garbage collection that uses an LRU policy that only reclaims entries that have not been used in some time (e.g, ten minutes or more) should be adequate.

A node should retain entries in the Default Router List and the Prefix List until their lifetimes expire. However, a node may garbage collect entries prematurely if it is low on memory. If doing so the node should retain at least one entry in the Default Router List (and preferably more than one) in order to maintain robust connectivity for off-link destinations.

When removing an entry from the Default Router List or the Prefix List there is no need to purge any entries from the Next-Hop or Neighbor Caches. Once again, Neighbor Unreachability Detection will effectively purge any entries in these caches that have become stale.

5. ROUTER AND PREFIX DISCOVERY

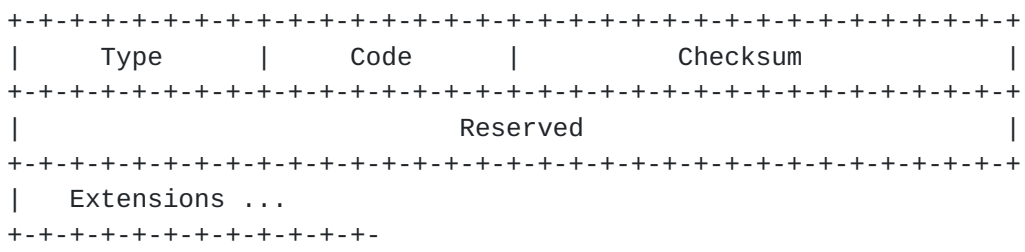
This section describes message formats, router behavior and host behavior related to the Router Discovery portion of Neighbor Discovery. Router Discovery is used to locate neighboring routers as well as learn prefixes and configuration parameters related to address autoconfiguration.

Prefix Discovery provides a mechanism through which hosts learn of ranges of IP addresses that reside on-link and thus can be reached directly without going through a router. Routers advertise a set of prefixes that cover those IP addresses that are on-link. Prefix discovery is logically separate from Router Discovery. In practice, prefix information is included in extension piggybacked on Router Advertisement messages to reduce network traffic.

Address Autoconfiguration information is also logically separate from Router Discovery. To reduce network traffic, however, autoconfiguration information is piggybacked on Router Discovery messages. This document does not define how autoconfiguration information is processed. See [ADDRCONF] for details.

5.1. Message Formats

5.1.1. Router Solicitation Message Format



IPv6 Fields:

Source Address
An IP address belonging to the interface from which this message is sent.

Destination Address
The all-routers multicast address.

Hop Count 1

Authentication Header
If a security association exists between the sender and the destination the sender SHOULD include this header.

IPv6 ICMP Fields:

Type 133

Code 0

Checksum The ICMPv6 checksum. See [ICMPv6].

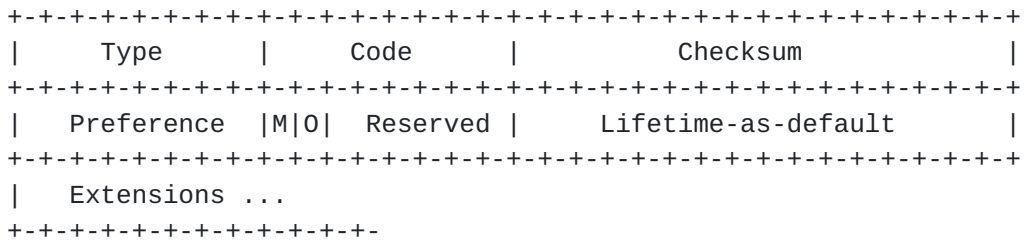
Reserved This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Extensions:

Source link-layer address
The link-layer address for the sender. SHOULD be included on link layers that have addresses in order for the router to be able to send a Router Advertisement without having to resolve the host's address.

Future versions of this protocol may define new extension types. Receivers MUST skip over and ignore any extensions they do not recognize and continue processing the message.

5.1.2. Router Advertisement Message Format



IPv6 Fields:

Source Address
: The interface's designated IP address.

Destination Address
: Either the Source Address of an invoking Router Solicitation or the all-hosts multicast address.

Hop Count
: 1

Authentication Header
: If a security association exists between the sender and the destination the sender SHOULD include this header.

IPv6 ICMP Fields:

Type
: 134

Code
: 0

Checksum
: The ICMPv6 checksum. See [ICMPv6].

Preference
: 8-bit unsigned integer. The preference as a default router. Larger numbers indicate more preferable routers.

M
: 1-bit flag. Use the administered (stateful) protocol for address autoconfiguration. The use of this flag is described in [ADDRCONF].

O
: 1-bit flag. Use the administered (stateful) protocol for autoconfiguration of other (non-address) information. The use of this flag is described in [ADDRCONF].

Reserved
: A 6-bit unused field. It MUST be initialized to zero

by the sender and ignored by the receiver.

Lifetime-as-default
16-bit unsigned integer. The lifetime associated with the default router in units of seconds. The maximum value corresponds to 18.2 hours. This lifetime does not apply to information contained in any extensions in the message. Extensions that need time limits for their information include their own lifetime fields.

Extensions:

Source link-layer address
The link-layer address for the router. Only used on link layers that have addresses. A router MAY omit this extension in order to enable inbound load sharing across multiple link-layer addresses.

Suggested hop limit
MAY be sent.

Suggested Neighbor Unreachability Timer
MAY be sent.

MTU SHOULD be sent on links that do not have a well-defined MTU. MAY be sent on links with a well-defined, standard MTU.

Prefix Information
A router MAY include 0 or more Prefix Information extensions. These extensions specify the prefixes that are on-link and also contain information specific to automatic address configuration. A router SHOULD include all on-link prefixes on multicast links. This enables multihomed hosts to do optimal outgoing interface selection for neighboring nodes.

Future versions of this protocol may define new extension types. Receivers MUST skip over and ignore any extensions they do not recognize and continue processing the message.

5.2. Router Specification

5.2.1. Router Configuration Variables

A router MUST allow for the following variables to be configured by system management; default values are specified so as to make it unnecessary to configure any of these variables in many cases.

For each multicast interface:

MaxRtrAdvInterval
: The maximum time allowed between sending multicast Router Advertisements from the interface, in seconds. MUST be no less than 4 seconds and no greater than 1800 seconds.

  Default: 600 seconds

MinRtrAdvInterval
: The minimum time allowed between sending unsolicited multicast Router Advertisements from the interface, in seconds. MUST be no less than 3 seconds and no greater than MaxRtrAdvInterval.

  Default: 0.75 * MaxRtrAdvInterval

RtrAdvLifetime
: The value to be placed in the Lifetime-as-default field of Router Advertisements sent from the interface, in seconds. MUST be no less than MaxRtrAdvInterval and no greater than 9000 seconds.

  Default: 3 * MaxRtrAdvInterval

PrefixList
: A list of prefixes to be placed in Prefix Information extensions in Router Advertisement messages sent from the interface.

  Default: The PrefixList contains all prefixes that the router advertises via routing protocols as being on the link on which the advertisement is sent.

  Each prefix is associated with:

  InvalidationLifetime
  : The value to be placed in the Invalidation Lifetime in the Prefix Information extension, in seconds.

Default: 3600 seconds.

On-link flag The value to be placed in the on-link flag in the Prefix Information extension.

Default: true.

Note: [ADDRCONF] defines additional information associated with the prefixes.

AdvertiseDefault
A flag indicating whether or not the router should advertise itself as a default router on the interface.

Default: TRUE

PreferenceLevel
The preferability of the router as a default router, relative to other routers on the same link. A 8-bit unsigned integer, with higher values meaning more preferable.

Default: 128

Designated address
The address to be used as the source address in all Neighbor Discovery messages sent on the interface.

Protocol constants are defined in section 10.

5.2.2. Message Validation by Routers

A router MUST silently discard any received Router Solicitation messages that do not satisfy the following validity checks:

- ICMP Checksum is valid.
- ICMP Code is 0.
- ICMP length (derived from the IP length) is 8 or more octets.
- if the message includes an Authentication Header, the message is correctly authenticated.

- all included extensions have a length that is greater then zero.

The contents of the Reserved field, and of any unrecognized extensions, MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new extensions; backward-incompatible changes may use different Code values.

A solicitation that passes the validity checks is called a "valid solicitation".

A router MAY silently discard any received Router Advertisement messages. It is recommended (but not required) that routers receive Router Advertisements and check for different MTU extension values and log a network management event when there is a mismatch. Routers can also examine the source address of Router Advertisements to determine which of a neighboring routers addresses is its designated address. Any other action on reception of such messages by a router is beyond the scope of this document.

5.2.3. Router Behavior

A router MUST join the all-router multicast address on all multicast capable interfaces.

The term "advertising interface" refers to any functioning and enabled interface that has at least one IP address assigned to it. From each advertising interface, the router transmits periodic, multicast Router Advertisements, containing the following values consistent with the message format above:

- In the Lifetime-as-default field: the interface's configured RtrAdvLifetime.

- In the Preference field: the interface's configured PreferenceLevel.

- In the O and M flags: see [ADDRCONF] for the settings of these flags.

- In the extension fields:

  Source Link-Layer Address extension: link-layer address of the sending interface. This extension MAY be omitted to facilitate in-bound load balancing on replicated interfaces.

  Prefix Information extensions: one Prefix Information extension for

each prefix listed in PrefixList. The "on-link" flag in the extension SHOULD be set to the on-link flag in the PrefixList entry. The Invalidation Lifetime in the extension is set to the InvalidationLifetime in the PrefixList entry. The use of the "address configuration flag" and the Deprecation Lifetime in the Prefix Information extension is specied in [ADDRCONF].

The advertisements are not strictly periodic: the interval between subsequent transmissions is randomized to reduce the probability of synchronization with the advertisements from other routers on the same link. This is done by maintaining a separate transmission interval timer for each advertising interface. Each time a multicast advertisement is sent from an interface, that interface's timer is reset to a uniformly-distributed random value between the interface's configured MinRtrAdvInterval and MaxRtrAdvInterval; expiration of the timer causes the next advertisement to be sent from the interface, and a new random value to be chosen. (It is recommended that routers include some unique value, such as one of their IP or link-layer addresses, in the seed used to initialize their pseudo-random number generators. Although the randomization range is configured in units of seconds, the actual randomly-chosen values SHOULD not be in units of whole seconds, but rather in units of the highest available timer resolution.)

For the first few advertisements sent from an interface (up to MAX_INITIAL_RTR_ADVERTISEMENTS), if the randomly chosen interval is greater than MAX_INITIAL_RTR_ADVERT_INTERVAL, the timer SHOULD be set to MAX_INITIAL_RTR_ADVERT_INTERVAL instead. Using this smaller interval for the initial advertisements increases the likelihood of a router being discovered quickly when it first becomes available, in the presence of possible packet loss.

In addition to the periodic, unsolicited advertisements, a router sends advertisements in response to valid solicitations received on any of its advertising interfaces. A router MAY choose to unicast the response directly to the soliciting host's address, or multicast it to the all-hosts address; in the latter case, the interface's interval timer is reset to a new random value, as with unsolicited advertisements. A unicast response MAY be delayed, and a multicast response MUST be delayed, for a small random interval not greater than MAX_RTR_RESPONSE_DELAY, in order to prevent synchronization with other responding routers, and to allow multiple, closely-spaced solicitations to be answered with a single multicast advertisement. While the router is delaying a multicast response it SHOULD silently ignore any additional solicitations, since it will multicast the response to all-hosts.

When a router receives a Router Solicitation it records the source of the packet as being a neighbor. If the solicitation contains a Source

Link-Layer Address extension the link-layer address is also recorded in the Neighbor Cache while also setting the "is_router" flag to false in the cache entry.

It should be noted that an interface may become an advertising interface at times other than system startup, as a result of recovery from an interface failure or through actions of system management such as:

- enabling the interface, if it had been administratively disabled, and its AdvertiseDefault flag is TRUE, or

- enabling IP forwarding capability (i.e., changing the system from being a host to being a router), when the interface's AdvertiseDefault flag is TRUE, or

- changing the AdvertiseDefault flag from FALSE to TRUE.

In such cases the router MUST commence transmission of periodic advertisements on the new advertising interface, limiting the first few advertisements to intervals no greater than MAX_INITIAL_RTR_ADVERT_INTERVAL. In the case of a host becoming a router, the system MUST also join the all-routers IP multicast group on all interfaces on which the router supports IP multicast (whether or not they are advertising interfaces).

An interface may also cease to be an advertising interface, through actions of system management such as:

- administratively disabling the interface, or

- shutting down the system, or disabling the IP forwarding capability (i.e., changing the system from being a router to being a host), or

- setting the AdvertiseDefault flag of the interface to FALSE.

In such cases the router SHOULD transmit a final multicast Router Advertisement on the interface with a Lifetime-as-default field of zero. In the case of a router becoming a host, the system MUST also depart from the all-routers IP multicast group on all interfaces on which the router supports IP multicast (whether or not they had been advertising interfaces). In addition, the host MUST insure that subsequent Neighbor Advertisement messages sent from the interface contain a Code of 0 (i.e. "not a router").

A router might want to send Router Advertisements without advertising itself as being a default router. For instance, a router might advertise prefixes for address autoconfiguration while not wishing to forward packets. Such a router SHOULD set the Lifetime-as-default field

to zero in its advertisements.

5.2.4. Designated Addresses

Routers should take some care in selecting their designated address and in handling any, hopefully infrequent, change of their designated address.

The designated address SHOULD be one that changes infrequently over time. Nodes receiving Neighbor Discovery messages use the source address to identify the sender. If multiple packets from the same neighbor contain different source addresses, nodes will assume they come from different nodes, leading to undesirable behavior. For example, a node will ignore Redirect messages that are believed to have been sent by a router other than the current first-hop router.

It is suggested that a link-local address be used as the designated address since this address does not change when a site renumbers.

If a router needs to change its designated address for one of its interfaces it SHOULD inform hosts of this change. The router should multicast a few Router Advertisements with lifetime-as-default set to zero for the old designated address and also multicast a few Router Advertisements for the new designated address. The exact procedures SHOULD be the same as when an interface ceases to being an advertising interface, and when an interface becomes an advertising interface, respectively.

A router MUST be able to determine the designated address for each of its neighboring routers in order to ensure that the target address in a Redirect message identifies the neighbor router by its designated address. This requires that routing protocols exchange designated addresses.

5.3. Host Specification

5.3.1. Host Configuration Variables

None.

5.3.2. Message Validation by Hosts

A host MUST silently discard any received Router Advertisement messages that do not satisfy the following validity checks:

- ICMP Checksum is valid.

- ICMP Code is 0.

- ICMP length (derived from the IP length) is 8 or more octets.

- if the message includes an Authentication Header, the message is correctly authenticated.

- all included extensions have a length that is greater then zero.

The contents of the Reserved field, and of any unrecognized extensions, MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new extensions; backward-incompatible changes may use different Code values.

An advertisement that passes the validity checks is called a "valid advertisement".

A host MUST silently discard any received Router Solicitation messages.

5.3.3. Host Behavior

The host joins the all-host multicast address on all multicast capable interfaces.

A host MUST NOT send a Router Advertisement message at any time.

To process a valid Router Advertisement, a host extracts the source address of the packet and does the following:

- If the address is not already present in the host's Default Router List, a new entry is added to the list. The entry's preference level is copied from the Preference field and a timer is started initialized to the advertisement's Lifetime-as-default field.

- If the address is already present in the host's Default Router List as a result of a previously-received advertisement, its preference level is updated and its timer is reset to the Lifetime-as-default value in the newly-received advertisement.

- If the received Lifetime-as-default value is zero the entry is timed out immediately.

After updating the Default Router List, the Router Advertisement MUST be

scanned for valid extensions. If the advertisement contains a source link-layer address extension the link-layer address MUST be recorded in the Neighbor Cache entry for the router and the "is_router" flag in the Neighbor Cache entry be set to true. This flag is used by Neighbor Unreachability Detection to determine when a router changes to being a host (i.e. no longer capable of forwarding packets).

For each Prefix Information extension that has the "on-link" (L) flag set, the host does the following:

- If the prefix is not already present in the Prefix List, create a new entry for the prefix and initialize its invalidation timer to the Invalidation Lifetime value in the Prefix Information extension.

- If the prefix is already present in the host's Prefix List as the result of a previously-received advertisement, reset its invalidation timer to the Invalidation Lifetime value in the Prefix Information extension.

- If the received Invalidation Lifetime value is zero the prefix is timed out immediately.

Whenever the invalidation timer expires for a Prefix List entry, that entry is discarded. No existing Next-Hop Cache entries are affected, however.

Whenever a timer expires for an entry in the Default Router List, that entry is discarded. Any entries in the Next-Hop Cache going through that router will continue to be used. Neighbor Unreachability Detection will purge them if appropriate.

To limit the storage needed for the Default Router List, a host MAY choose not to store all of the router addresses discovered via advertisements. If so, the host SHOULD discard those addresses with lower preference levels in favor of those with higher levels. In any case, a host SHOULD retain more than one default router address in the list so that, if the current choice of default router is discovered to be down, the host can immediately select another default router, without having to wait for the next advertisement to arrive.

Preference levels learned from advertisements do not affect any of the host's cached route entries. For example, if the host has been redirected to use a particular router address to reach a specific IP destination, it continues to use that router address for that destination, even if it discovers another router address with a higher preference level. Preference levels influence the choice of router only

for an IP destination for which there is no Next-Hop Cache entry, or whose Next-Hop Cache entry points to a router that is subsequently discovered to be unreachable.

The algorithm for selecting a router depends in part on whether or not a router is known to be reachable. The exact details of how a node keeps track of a neighbor's reachability status are covered in Section 6.3. The algorithm for selecting a default router is invoked only when a Next-Hop Cache entry is incomplete or when communication through an existing router appears to be failing. Under normal conditions, a router would be selected the first time traffic is sent to a destination, with subsequent traffic for that destination using the same router as indicated in the Next-Hop Cache. The policy for selecting routers from the Default Router List is as follows:

1) Examine router entries one at a time by iterating through the list in preference order, starting with the highest preference router. For each examined entry in the list:

   a) If the status of the Neighbor Cache entry for the router is REACHABLE or PROBE, select this router. Otherwise, continue to the next step.

   b) If the recorded status of the neighboring router is TRY_ALTERNATES, the router MUST NOT be selected for use. However, the Neighbor Unreachability Detection mechanism (see section 6.3) SHOULD be invoked to send a (rate limited) probe to the router in order to solicit a reachability confirmation.

2) If the entire router list is scanned in 1) without finding an acceptable candidate, the host MAY either send an ICMP Destination Unreachable error (as specified in [ICMPv6], or it MAY assume that the destination is on-link. The latter is likely to result in invoking Address Resolution for the destination, which may in turn result in generating an ICMP Address Unreachable error if the destination fails to respond to repeated Neighbor Solicitation messages.

The above algorithm has a number of desirable properties. First, higher preference routers that are known to be reachable are favored over those having lower preferences. However, higher preference routers are not favored to the point that routers considered to be unreachable continue to be chosen when other candidates are available. Second, when all routers on the default list are considered unreachable, all candidate routers are probed. Finally, when higher-preference routers are unreachable, but lower-preference routers are reachable, the higher-preference routers are still periodically probed, and they will become candidates for selection as soon as they become reachable again.

A host is permitted (but not required) to transmit up to MAX_RTR_SOLICITATIONS Router Solicitation messages from any of its multicast interfaces after any of the following events:

- The interface is initialized at system startup time.
- The interface is reinitialized after a temporary interface failure or after being temporarily disabled by system management.
- The system changes from being a router to being a host, by having its IP forwarding capability turned off by system management.
- The host is re-attached to a link after being detached for some time.

The IP destination address of the solicitations is the all-routers multicast address. The IP source address contains one of the interface's addresses. The Source Link-Layer Address extension is set to the host's link-layer address.

If a host does choose to send a solicitation after one of the above events, it SHOULD delay that transmission for a random amount of time between 0 and MAX_RTR_SOLICITATION_DELAY. This serves to alleviate congestion when many hosts start up on a link at the same time, such as might happen after recovery from a power failure. (It is recommended that hosts include some unique value, such as one of their IP or link-layer addresses, in the seed used to initialize their pseudo-random number generators. Although the randomization range is specified in units of seconds, the actual randomly-chosen value SHOULD not be in units of whole seconds, but rather in units of the highest available timer resolution.)

A host MAY also choose to further postpone its solicitations, subsequent to one of the above events, until the first time it needs to use a default router.

Upon receiving a valid advertisement the host MUST desist from sending any solicitations on that interface (even if none have been sent yet), until the next time one of the above events occurs. The small number of retransmissions of a solicitation, which are permitted if no such advertisement is received, SHOULD be sent at intervals of RTR_SOLICITATION_INTERVAL seconds, without randomization.

6. ADDRESS RESOLUTION AND NEIGHBOR UNREACHABILITY DETECTION

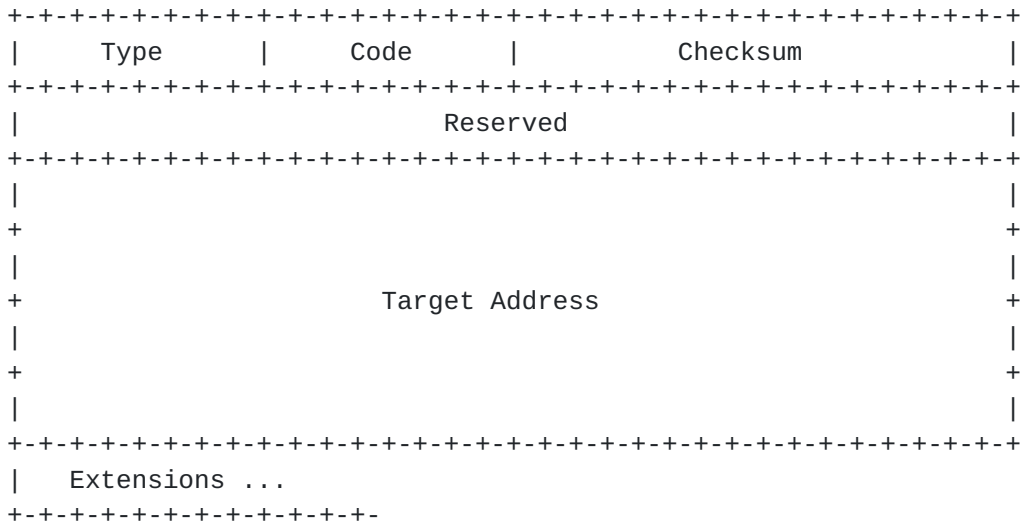
This section describes the set of functionality related to the Neighbor Solicitation and Neighbor Advertisement messages and includes descriptions of the Address Resolution and the Neighbor Unreachability Detection algorithms.

These messages are also used for Duplicate Address Detection as specified by [ADDRCONF]. In particular, Duplicate Address Detection uses the unspecified address as the Source Address in Neighbor Solicitations to prompt a node with a duplicate address to multicast the Neighbor Advertisement.

6.1. Message Formats

6.1.1. Neighbor Solicitation Message Format

Nodes send Neighbor Solicitations to request the link-layer address of a target node while 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.



IPv6 Fields:

Source Address

An IP address belonging to the interface from which this message is sent. If the sender is a router, the address MUST be the interface's designated address. A

node MAY also use the unspecified address before it has determined that its addresses are unique.

Destination Address
Either the solicited-node multicast address corresponding to the target address, or the target address.

Hop Count 1

Authentication Header
If a security association exists between the sender and the destination the sender SHOULD include this header.

IPv6 ICMP Fields:

Type 135

Code

0	If the sender is a host.
1	If the sender is a router.

Checksum The ICMPv6 checksum. See [ICMPv6].

Reserved This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Target Address The IP address of the target of the invoking solicitation or, for an unsolicited advertisement

Extensions:

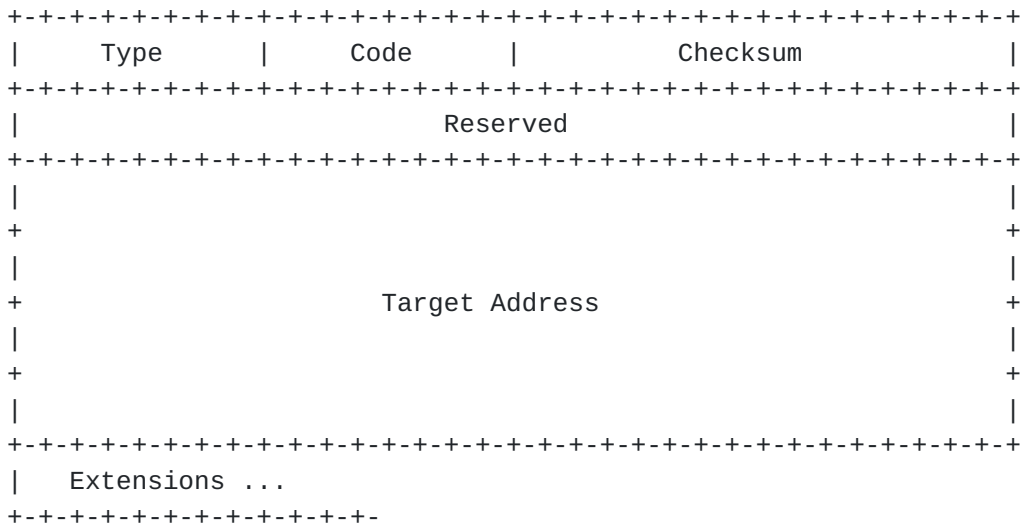
Source link-layer address
The link-layer address for the sender. MUST be included on link layers that have addresses.

Future versions of this protocol may define new extension types. Receivers MUST skip over and ignore any extensions they do not recognize and continue processing the message.

6.1.2. Neighbor Advertisement Message Format

A node MUST send a Neighbor Advertisement in response to a Neighbor Solicitation for an IP addresses assigned to the receiving interface. In addition a node MAY send an unsolicited multicast Neighbor

Advertisement when the node knows that its link-layer address has changed.



IPv6 Fields:

Source Address
: An IP address belonging to the interface from which this message is sent. The source address MUST be the same as the target address for a non-proxy response. The source address MUST be the interface's designated address for a proxy response.

Destination Address
: Either the Source Address of an invoking Neighbor Solicitation, or the all-nodes multicast address. If the source solicitation is the unspecified address the advertisement MUST be multicast to all-nodes.

Hop Count
: 1

Authentication Header
: If a security association exists between the sender and the destination the sender SHOULD include this header.

IPv6 ICMP Fields:

Type
: 136

Code
: 0 If the sender is a host.

1 If the sender is a router.

Checksum The ICMPv6 checksum. See [ICMPv6].

Reserved This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Target Address The address from the Target Address field in the Neighbor Solicitation message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address has changed.

Extensions:

Target link-layer address
The link-layer address for the target. MUST be included on link layers that have addresses.

Future versions of this protocol may define new extension types. Receivers MUST skip over and ignore any extensions they do not recognize and continue processing the message.

6.2. Address Resolution

Address Resolution provides the mechanism through which nodes determine the link-layer address of their neighbors.

6.2.1. Message Validation by Nodes

A node MUST silently discard any received Neighbor Solicitation or Advertisement messages that do not satisfy the following validity checks:

- ICMP Checksum is valid.
- ICMP Code is 0 or 1.
- ICMP length (derived from the IP length) is 24 or more octets.
- if the message includes an Authentication Header, the message is correctly authenticated.
- all included extensions have a length that is greater then zero.

The contents of the Reserved field, and of any unrecognized extensions,

MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new extensions; backward-incompatible changes may use different Code values.

Neighbor Solicitations and Advertisements that passes the validity checks are called "valid solicitations" and "valid advertisements", respectively.

6.2.2. Node Specification

When a multicast-capable interface is initialized 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.

6.2.3. Sending Node Specification

When a node has a packet to send, but does not know the next-hop's link-layer address, the sender performs address resolution by transmitting a Neighbor Solicitation message targeted at the neighbor and queuing the packet. The message MUST be sent to the solicited-node multicast address corresponding to the target address.

The sender MUST include its link-layer address (if it has one) in the solicitation as a Source Link-Layer Address extension, so that the receiver discovers the sender's link-layer address without the need for an additional packet exchange.

While waiting for address resolution to complete, the sender MUST maintain a small queue containing packets waiting for address resolution to complete. The queue MUST hold at least one packet, and MAY contain more. However, the number of queued packets per neighbor SHOULD be limited to some small value. When a queue overflows, the new arrival SHOULD replace the oldest entry. Once address resolution completes, all queued packets SHOULD be transmitted.

While awaiting for address resolution to complete, the sender MUST rate-limit the sending of further Neighbor Solicitations to the neighbor to at most one solicitation every RESOLVE_RETRANS_TIMER seconds. This constraint applies even if the sender has new packets to send to the neighbor at a higher rate.

In order to be able to generate ICMP Address Unreachable errors, the sender SHOULD retransmit the solicitation every RESOLVE_RETRANS_TIMER seconds until either an advertisement is received from the target or the solicitation has been retransmitted UNREACHABLE_THRESHOLD times.

If no Neighbor Advertisement is received after sending UNREACHABLE_THRESHOLD unanswered solicitations, the sender SHOULD generate an ICMP unreachable error with code 3 (Address Unreachable) for each packet queued for the neighbor. The error messages are constructed as all ICMP errors (see [ICMPv6]) and sent errors to the sources of the queued packets. Generating ICMP errors when address resolution fails provides more precise diagnostics to administrators which is the intent of the Address Unreachable code in [ICMPv6].

When a valid unicast Neighbor Advertisement is received, and there is a Neighbor Cache entry for the target which contains no link-layer address, the node records the link-layer address in the Neighbor Cache entry and also sends any packet that have been queued for the neighbor. Furthermore, the node MUST set the "is_router" flag in the Neighbor Cache entry based on the Code field in the advertisement. If the "is_router" flag was previously set but the advertisement has Code set to 0 the node MUST follow the rules in section 6.3.2 to handle the case when a router becomes a host.

Multiple unicast Neighbor Advertisements can be received in response to a query. In such cases one or more of the advertisements is a proxy advertisement. Proxy advertisements are identified by having differing source and target addresses. A node MUST give preference to non-proxy responses over proxy responses and, among multiple proxy responses, a node MUST prefer the first proxy response. This is accomplished by applying the following rules while processing received advertisements:

- if no link-layer address has been previously recorded, install the one contained in the advertisement.

- if a link-layer address has already been recorded, and the advertisement is not a proxy advertisement, install the address contained in the advertisement.

- otherwise ignore the advertisement

A node MAY occasionally multicast unsolicited Neighbor Advertisement announcing a link-layer address change. A node that receives an multicast Neighbor Advertisement does the following:

- It MUST silently ignore a proxy multicast Neighbor Advertisement.

- If the node does not have a Neighbor Cache entry for the target of the advertisement, it SHOULD silently discard the message. Accepting such multicast advertisements would result in occupying a cache entry with information about a neighbor that might never be used.

- If the node does have a Neighbor Cache entry for the target, it SHOULD copy the link-layer address information contained in the advertisement's Source Link-Layer Address extension into the corresponding Neighbor Cache entry.

- The node MUST not treat the receipt of a multicast advertisement as a confirmation that the neighbor is REACHABLE (as defined in Section 4.1). See section 6.3.1.

6.2.4. Target Node Specification

When a node receives a valid Neighbor Solicitation, it compares the query's Target Address against the IP addresses belonging to the incoming interface. If the node is a router it MUST also compare the Target Address against the set of anycast addresses (and potentially other addresses) for which it is providing proxy services. If no match is found the node is not the target of the query and it MUST silently ignore the solicitation.

If the node is the target of the solicitation, it first ensures that it has an up-to-date neighbor cache for the Source Address of the solicitation. If no entry is found one is created and its link-layer address is copied from the Source Link-Layer Address extension in the message. If an entry already exists its link-layer address is updated to match the address in the Source Link-Layer Address extension. In either case , the node MUST set the "is_router" flag in the Neighbor Cache entry based on the Code field in the solicitation. If the "is_router" flag was previously set but the advertisement has Code set to 0 the node MUST follow the rules in section 6.3.2 to handle the case when a router becomes a host.

If the source of the solicitation is the unspecified address, the target MUST multicast an advertisement to the all-nodes address. Otherwise, the target MUST send a unicast Neighbor Advertisement to the address copied from the IP Source Address of the Neighbor Solicitation. In both cases the Target Address is copied from the solicitation message to the advertisement and the Target Link-Layer Address extension is filled with the node's link-layer address on the link. If the node is not providing proxy services for the targeted address, the IP Source Address MUST be set to the address in the Target Address field (which is one of the IP addresses belonging to the interface). This guarantees that the receiver can identify the Neighbor Advertisement as being a non-proxy advertisement.

If the node is providing proxy services for the target the IP Source Address MUST be set the interface's designated address (which is

different than the Target Address). This allows the receiver to recognize the message as a proxy advertisement.

A node MUST NOT send unicast Neighbor Advertisement except in response to a Neighbor Solicitation, in order to avoid confusing the Neighbor Unreachability Detection algorithm.

6.2.5. Anticipated link-layer address changes

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 up to MAX_NEIGHBOR_ADVERTISEMENT Neighbor Advertisement messages to the all-nodes multicast address. These advertisements MUST be separated by at least MIN_NEIGHBOR_ADVERT_INTERVAL seconds.

The Target Address field in the multicast advertisement is set to the IP address of the interface and the Target Link-Layer Address extension is filled with the new link-layer address. The IP Source Address MUST match the address in the Target Address field of the solicitation.

A node that has multiple IP addresses assigned to an interface MAY multicast a separate Neighbor Advertisement for each address.

A proxy MUST NOT multicast Neighbor Advertisements when its link-layer changes. (It is anticipated that multiple routers will proxy for the same addresses and allowing multicast advertisement could result in excessive multicast traffic.)

Note that multicasting Neighbor Advertisements does not reliably update caches in all nodes (the advertisements might not be received by all nodes) and should only be viewed as a optimization to quickly update the caches in most neighbors. The Neighbor Unreachability Detection algorithm will ensure that neighbors reliably update the cached link-layer address when they attempt to communicate with the node.

6.2.6. Proxy Neighbor Advertisements

Under limited circumstances, a router MAY proxy for one ore 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 may accept packets addressed to one of its configured anycast addresses, or a router might potentially accept packets on behalf of a mobile node that has moved off-link. The address being served is called a "proxee" in this section.

A proxy MUST join the solicited-node multicast address(es) that correspond to the proxee's IP address(es).

All proxy Neighbor Advertisement messages MUST be tagged as being proxy messages; the advertisement's Source Address MUST differ from its Target Address (e.g., the proxee). In practice, this requirement poses no special burden. By definition, the advertisement's source address MUST be the designated address of the interface on which the advertisement is sent, which will be different than any proxee address.

6.2.7. Anycast

An anycast address can not be syntactically distinguished from other unicast addresses. This section shows how the rules defined above "do the right thing" for anycast addresses.

When a router responds to a Neighbor Solicitation for an anycast address, it by definition responds with a proxy Neighbor Advertisement. Anycast address are not permitted to appear as the source address in an IP packet, guaranteeing that the advertisement's source and target addresses differ.

A node might receive multiple Neighbor Advertisements in response to a Neighbor Solicitation for an anycast address when multiple neighbors are configured to recognize the anycast address. The precedence rules in section 6.2.3 will make the node select the first advertisement (i.e. the fastest or lightest loaded router) as current binding for the anycast address.

The use of Neighbor Unreachability Detection ensures that a node quickly detects when the current binding for the anycast address has gone stale e.g. due to a router no longer belonging to the anycast address.

6.3. 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, a mobile node moving off-link, etc. If the destination has failed, no recovery is possible and communication fails. On the other hand, if it is the path that has failed, recovery may be possible. Thus, a node actively tracks the reachability "state" for the neighbors to which it is sending packets.

Neighbor Unreachability Detection is used for all paths between

neighboring nodes, including host-to-host, host-to-router, and router-to-host communication. When a path to a neighbor appears to be failing, the specific recovery attempt depends on how the neighbor is being used. For example, appropriate recovery procedures when using the neighbor as a router differ from those appropriate for the case where the neighbor is the destination.

6.3.1. Reachability Confirmation

A neighbor is considered reachable if the node has recently received a confirmation that packets sent to the neighbor are received by its IPv6 layer. Positive confirmation can be gathered in two ways: hints from upper layer protocols that indicate a connection is making "forward progress", or receipt of a unicast Neighbor Advertisement message that is a response to an explicit Neighbor Solicitation probe.

A connection makes "forward progress" if the packets received from a remote peer can only be arriving if recent packets sent to that peer are actually reaching it. For example, receipt of a (new) acknowledgement indicates that previously sent data reached the peer. Likewise, the arrival of a new (non-duplicate) packet indicates that earlier acknowledgements are being delivered to the remote peer. If packets are reaching the peer the packets must also be reaching the sender's next-hop neighbor, thus "forward progress" is a confirmation that the next-hop neighbor is reachable. When available, this upper-layer information SHOULD be used.

In some cases (e.g, UDP-based protocols and routers forwarding packets to hosts) such reachability information is not available from upper-layer protocols. When no hints are available and a node is sending packets to a neighbor, the node actively probes the neighbor using Neighbor Solicitation messages to verify that the forward path is still working.

The receipt of a unicast Neighbor Advertisement that is a response to such a Neighbor Solicitation probe serves as a reachability confirmation, since all unicast advertisements are sent in response to a solicitation. A received multicast Neighbor Advertisement MUST NOT be treated as a reachability confirmation since it is likely to be unsolicited. Receipt of unsolicited advertisements only confirm the one-way path from the neighbor to the recipient node. In contrast, Neighbor Unreachability Detection requires that a path be working from the node to the neighbor. An advertisement sent in response to an explicit solicitation confirms that a path is working in both directions; the solicitation reached the neighbor, prompting it to generate an advertisement, and the advertisement reached the querying node.

6.3.2. Node behavior

Neighbor Unreachability Detection operates in parallel with the sending of packets to a neighbor. While reasserting a neighbor's reachability, a node continues sending packets to that neighbor using the cached link-layer address.

A neighbor is considered REACHABLE if a reachability confirmation was received less than REACHABLE_TIME seconds ago. Packets sent to a REACHABLE neighbor require no special action.

Neighbors with PROBE or TRY_ALTERNATES status are actively probed to ascertain their reachability status. Neighbor Solicitation probe messages are sent only on demand; only when a packet is being sent to that neighbor. When no traffic is sent to a neighbor, no probes are sent to it, regardless of the neighbors reachability state.

When a REACHABLE Neighbor Cache entry is referenced after REACHABLE_TIME seconds have passed since the last reachability confirmation was received, its status should be changed to PROBE but no probe should be sent. Any probing is deferred for an additional DELAY_FIRST_PROBE_TIME seconds; an optimization that gives the upper-layer protocol additional time to provide a reachability confirmation in those cases where REACHABLE_TIME seconds have passed since the last confirmation due to lack of recent traffic. Without this optimization the opening of a TCP connection after a traffic lull would initiate probes even though the subsequent three-way handshake would provide a reachability confirmation almost immediately.

Probe messages are rate limited. Consecutive probe messages to the same neighbor MUST be separated by a delay of at least REACHABLE_RETRANS_TIME seconds. The actual inter-probe delay depends on the traffic pattern; probe MUST be sent when a packet is sent to the neighbor and REACHABLE_RETRANS_TIME seconds has passed since sending the previous probe.

Probe messages sent while in PROBE status are unicast to the neighbor using the cached link-layer address. Probes that are sent in TRY_ALTERNATES state are multicast to the solicited-node address just like regular Neighbor Solicitations are when resolving the link-layer address.

After CONSECUTIVE_UNICAST_PROBES probes have been sent without receiving any reachability confirmation, the neighbor state should be changed from PROBE status to TRY_ALTERNATES and the node should attempt to find an alternate path. This is accomplished by discarding the cached link-layer address and invoking the next-hop determination procedure (described in Section 4.2) for the packet. If the next-hop is being

used as a router, performing the next-hop calculation may result in selecting another default router. If the destination was thought to be on-link, but the set of on-link prefixes has changed, recalculating the next-hop may result in a switch to a router. In other cases the next-hop determination might find that the neighbor is still presumed to be on-link in which case the regular Address Resolution mechanism will be invoked; that mechanism will then multicast Neighbor Solicitations to the neighbor.

If a packet is about to be sent to a neighbor whose status is already TRY_ALTERNATES, the packet should not be sent. Instead the next-hop determination should be invoked for the destination in order to select a different next-hop as above. This case occurs when multiple Next-Hop Cache entries refer to the same Neighbor Cache entry and the use of one of the next-hop entries has previously resulted in transitioning to TRY_ALTERNATES status. In this case other next-hops using the same neighbor should attempt to find an alternate path immediately when sending the next packet.

In addition to being used when sending packets to a neighbor, Neighbor Unreachability Detection is also invoked by the default router selection policy in section 5.3.3 to send a probe message without actually sending a data packet. In this case the reachability status is TRY_ALTERNATES and the node should multicast a Neighbor Solicitation to the solicited-node address as an attempt to receive a reachability confirmation for the default router.

To detect a router that switches from being a router to being a host (e.g, by having its IP forwarding capability turned off by system management), a node MUST compare the Code field of all received Neighbor Advertisement messages with the "is_router" flag recorded in the Neighbor Cache entry. When a node detects that a neighbor has changed from being a router to being a host, the node MUST remove that router from the Default Router List and update the Next-Hop Cache so that all entries using that neighbor as a router switch to another router. Note that a router may not be listed in the Default Router List, but still have Next-Hop Cache entries using it, if a host was redirected to it.

An algorithmic specification of the above mechanism is presented in section 6.3.4.

6.3.3. Reachability State

For the purpose of describing the Neighbor Unreachability Detection algorithm, this document uses the following state-related variables for each neighbor:

- the number of consecutive unanswered Neighbor Solicitation probes.

- a "time-of-next-event" event timer that specifies when some action must be taken. Note that in contrast to timers used by e.g. transport protocols for scheduling retransmissions, this timer does not trigger event processing at the time at which it expires. Instead, it is examined only when a packet is being transmitted to the neighbor. This on-demand event processing can be implemented by comparing the current time with the time-of-next-event whenever a neighbor entry is referenced while sending a packet.

- a status variable that take one of the values REACHABLE, PROBE, or TRY_ALTERNATES as defined informally in section 4.1. This variable is primarily used to add clarity to the specification. An implementation might only need to keep track the number of unanswered probes and the time-of-next-event timer; they can be made to implicitly define the current status.

A node MUST track the above state on a per-neighbor basis. In particular, a node MUST maintain a single Neighbor Cache entry for a router even though many Next-Hop Cache entries might refer to the same router, in order to avoid redundant probing of the router.

6.3.4. Algorithm

When a node is confirmed reachable, its status is set to REACHABLE, its time-of-next-event is set to the current time plus REACHABLE_TIME and the count of consecutive unanswered probes is set to -1.

All other actions in Neighbor Unreachability Detection take place when sending or attempting to send packets to the neighbor. Note that no actions are triggered by an explicit timeout.

Whenever a packet is sent to a neighbor, the current time is compared to the time-of-next-event. If the time-of-next-event exceeds the current time, the node performs the following actions based on the current state:

1) If the status is REACHABLE, change the status to PROBE, set the number of unanswered probes to 0, set time-of-next-event to current time plus DELAY_FIRST_PROBE_TIME, and send the packet. No probe is sent. This is the optimization that defers the sending of any probe until the upper-layer has had a reasonable time to provide a reachability confirmation.

2) If the status is PROBE and less than CONSECUTIVE_UNICAST_PROBES

have been sent, send a unicast Neighbor Solicitation to the cached link-layer address, increment the number of probes, and send the packet. Set time-of-next-event to current time plus REACHABLE_RETRANS_TIME. This ensures that the next probe will not be sent until at least REACHABLE_RETRANS_TIME seconds have elapsed, rate-limiting consecutive probe messages for the neighbor to at most one message every REACHABLE_RETRANS_TIME seconds.

3) If the status is PROBE and CONSECUTIVE_UNICAST_PROBES have been sent, the neighbor is likely to be unreachable. Change the status to TRY_ALTERNATES, discard the cached link-layer address, and perform next-hop determination for the destination. The packet is then sent using the (potentially different) next hop that resulted from the next-hop determination.

In addition, when sending a packet the reachability state of the neighbor SHOULD be always checked, independently of the time-of-next-event, to be able to quickly perform next-hop determination when the status is TRY_ALTERNATES. When status is TRY_ALTERNATES a next-hop determination is always performed and the packet is then sent using the determined next-hop.

If the Neighbor Unreachability Detection is invoked from the default router selection policy (section 5.3.3) this check should be performed:

- If the status is TRY_ALTERNATES and time-of-next-event is exceeds the current time, then multicast a probe to the solicited-node multicast address corresponding to the neighbor's address, increment the number of probes, and set time-of-next-event to the current time plus DEFAULT_RTR_PROBE_INTERVAL. This will solicit a default router for a reachability confirmation at most every DEFAULT_RTR_PROBE_INTERVAL while a different, known to be reachable, default router is selected by the default router selection policy.

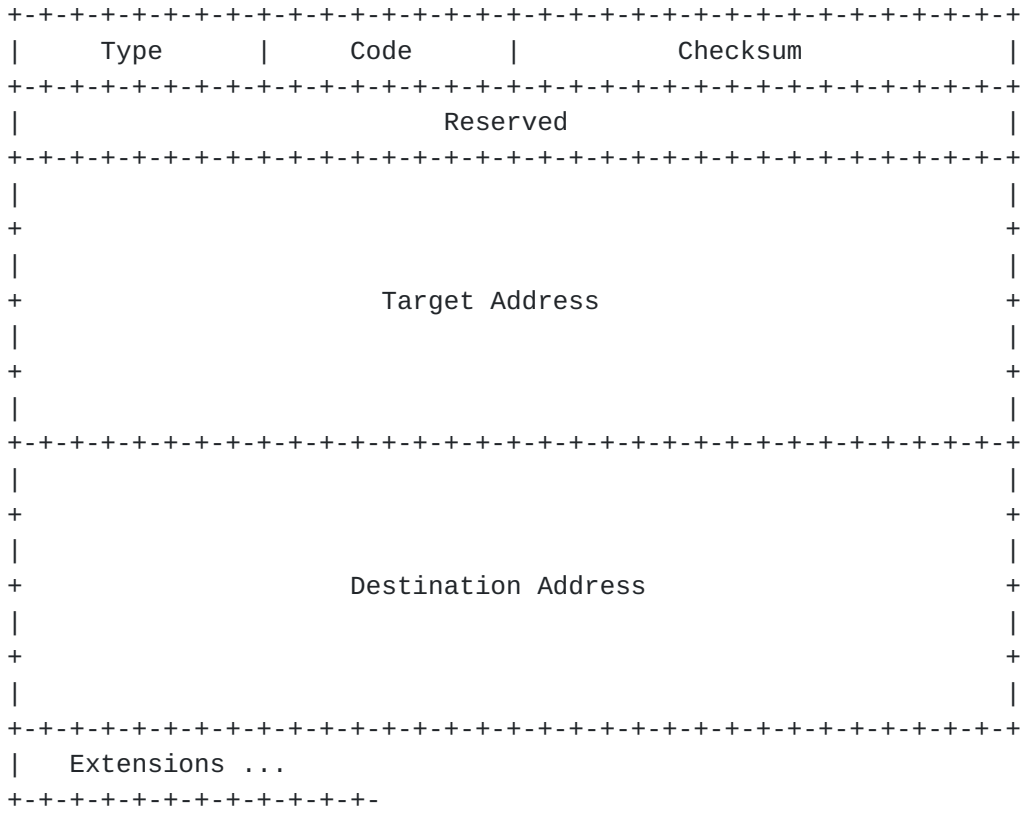
7. REDIRECT FUNCTION

This section describes the set of functionality related to the sending and processing of Redirect messages.

7.1. Redirect Message Format

A Redirect packet is sent from a router to a host to inform the host of

a better first-hop node on the path to a destination.



IPv6 Fields:

Source Address
The designated IP address of the interface from which this message is sent.

Destination Address
The Source Address of the packet that triggered the redirect.

Hop Count 1

Authentication Header
If a security association exists between the sender and the destination the sender SHOULD include this header.

IPv6 ICMP Fields:

Type 5

Code

0 If the target is a host.
1 If the target is a router.

Checksum The ICMPv6 checksum. See [ICMPv6].

Reserved This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Target Address An IP address of the node to which traffic for the Destination SHOULD be sent. When the target is a router this MUST be the router's designated address on the link. This is required so that hosts can uniquely identify the routers by their designated address.

Destination Address
The IP address of the destination which is redirected to the target.

Extensions:

Target link-layer address
The link-layer address for the target. It SHOULD be included on link layers that have addresses, if known.

Redirected Header
As much as possible of the IPv6 packet that triggered the sending of the Redirect without making the redirect packet exceed 576 octets.

Future versions of this protocol may define new extension types. Receivers MUST skip over and ignore any extensions they do not recognize and continue processing the message.

7.2. Router Specification

A router SHOULD send a redirect message, subject to rate limiting, whenever it forwards a packet in which:

- the Source Address field of the packet identifies a neighbor, and

- after consulting its routing table, the router forwards the packet to a node residing on the same link as the packet's source, and

- the Destination Address of the packet is not a multicast address, and

- the packet is not source routed through the router. A packet is

source routed through the router if, when the packet is received by the router, it contains the IPv6 route header and the router's address is in the Destination Address field.

The transmitted redirect packet contains, consistent with the above message format:

- In the ICMP Code field: set to 0 if the target is a host and 1 if it is a router.

- In the Target Address field: the address to which subsequent packets for the destination SHOULD be sent. If the target is a router this MUST be set to the target's designated address on the link.

- In the Destination Address field: the destination address of the invoking IP packet.

- In the extension fields:

  Target Link-Layer Address extension: link-layer address of the target, if known.

  Redirected Header: as much of the forwarded packet as can fit without the redirect packet exceeding 576 octets in size.

A router MUST limit the rate of Redirect messages sent, in order to limit the bandwidth and processing costs incurred by the Redirect messages when the source does not correctly respond to the Redirects, or the source chooses to ignore unauthenticated Redirect messages. Examples of how to implement such a rate-limiting function are in [ICMPv6].

A router MUST NOT update its routing tables upon receipt of a Redirect.

7.3. Host Specification

7.3.1. Message Validation by Hosts

A host MUST silently discard any received Redirect messages that do not satisfy the following validity checks:

- ICMP Checksum is valid.

- ICMP Code is 0 or 1.

- ICMP length (derived from the IP length) is 40 or more octets.
- the IP source address of the Redirect is the same as the current first-hop router for the specified destination.
- the Target Address of the redirect is not a multicast address.
- the Destination Address field in the redirect message does not contain a multicast address.
- if the message includes an Authentication Header, the message is correctly authenticated.
- all included extensions have a length that is greater then zero.

The contents of the Reserved field, and of any unrecognized extensions MUST be ignored. Future, backward-compatible changes to the protocol may specify the contents of the Reserved field or add new extensions; backward-incompatible changes may use different Code values.

A host MUST NOT consider a redirect invalid just because the Target Address of the redirect is not covered under one of the link's prefixes.

A redirect that passes the validity checks is called a "valid redirect".

7.3.2. Host Behavior

A host receiving a valid redirect SHOULD update its routing information accordingly. When a redirect is received the host updates the Next-Hop Cache entry for the destination to point to the target. If no Next-Hop Cache entry exists for the destination such an entry is created.

If the redirect contains a Target Link-Layer Address extension the host either creates or updates the Neighbor Cache entry for the target. The link-layer address in the Neighbor Cache entry MUST be copied from the Target Link-Layer Address extension. In addition, if the Code in the redirect is set to 1 the "is_router" flag is set to true in the Neighbor Cache entry. Otherwise the "is_router" flag SHOULD be set to false.

A host MAY ignore a Redirect message that does not have an IPv6 Authentication header.

A host MUST NOT send Redirect messages.

8. EXTENSIONS

Extensions provide a mechanism to encode variable length as well as optional pieces of information in the different ND packets.

Extensions can also be used to add additional functionality to ND. Examples of potential future functionality is better support for links with asymmetric connectivity and better support for NBMA links that use "address resolution servers" in IPv4.

In order to ensure this extensibility all nodes MUST skip over any extensions they do not recognize in received ND packets and continue processing the packet. However, the extensions specified in this document MUST be implemented by all implementations.

The current set of extensions are defined in order to allow receivers to process multiple extensions in the same packet independently of each other. In order to maintain these properties future extensions SHOULD follow the simple rule:

> The extension MUST NOT depend on the presence or absence of any other extensions. The semantics of an extension should depend only on the information in the fixed part of the ND packet and on the information contained in the extension itself.

This constraint allows receivers to process extensions independently (e.g., an implementation can choose to process the Prefix Information extension in a Router Advertisement message in a user-space process while the link-layer address in the same message is recorded by the kernel).

The constraint can also be useful should we ever need to send more extensions then can fit in a single packet; multiple packets can carry subsets of the extensions without any change in semantics.

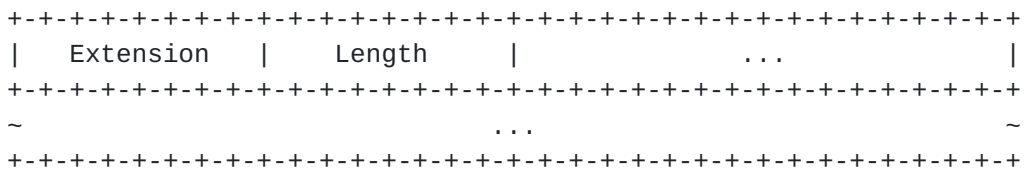
When multiple extensions are present in a Neighbor Discovery packet, they may appear in any order; receivers MUST be prepared to process them independently of their order.

Senders MAY send a subset of extensions in different packets. For instance, if the prefix Invalidation Lifetime is high it might not be necessary to include the Prefix Information extension in every Router Advertisement. In addition, different routers might send different sets of extensions. Thus, a receiver MUST NOT associate any action with the absence of an extension in a particular packet. This protocol specifies that receivers should only act on the expiration of timers and on the information that is received in the packets.

All extensions have a length that is a multiple of 8 octets. This makes it simple to ensure appropriate alignment without any "pad" extensions. The fields in the extensions, as well as the fields in the ND packets, are defined to align on their natural boundaries (e.g. a 16-bit field is aligned on a 16-bit boundary) except the 128-bit IP addresses/prefixes which are aligned on a 64-bit boundary.

The link-layer address field contains an octet string thus it is only aligned on an 8-bit boundary.

All extensions are of the form:



Fields:

Extension 8-bit identifier of the type of extension. The extensions defined in this document are:

Extension Name	Extension
Source Link-Layer Address	1
Target Link-Layer Address	2
Prefix Information	3
Redirected Header	4
Suggested Hop Limit	5
Neighbor Unreachability Detection Timer	6
MTU	7

Length 8-bit unsigned integer. The length of the extension in units of 8 octets. The value 0 is invalid. Nodes MUST silently discard an ND packet that contains an extension with length zero.

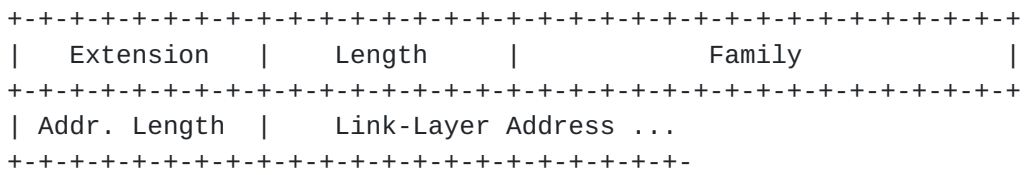
The size of an ND packet including the IP header is limited to the link MTU (which is at least 576 octets). When adding extensions to an ND packet a node MUST NOT exceed the link MTU. This is handled in a packet specific manner.

The only ND packets that currently can exceed the link MTU are Router Advertisements and Redirects; the former due a large number of Prefix Information extensions and the latter due to the Redirected Header extension.

If there are more Prefix Information extensions than can fit in a single Router Advertisement packet the router MUST send multiple separate advertisements that each contain a subset of the set of prefixes.

In a Redirect packet the amount of data included in the Redirected Header MUST be limited so that the packet does not exceed 576 octets.

8.1. Source/Target Link-layer Address



Fields:

Extension
: 1 for Source Link-layer Address
2 for Target Link-layer Address

Length
: The length of the extension in units of 8 octets. For example, the length with IEEE 802 addresses is 2.

Family
: The link-layer Address Family Number. Up-to-date values are specified in the most recent "Assigned Numbers RFC" [RFC-1700].

Addr. Length
: The length of the actual link-layer address. The unit for this length depends on the Address Family.

  The address length field is in units of octets except for those families for which it is in the unit of nibbles (4-bits):

  - E.163
  - E.164 (SMDS, Frame Relay, B-ISDN)
  - F.69 (Telex)
  - X.121 (X.25, Frame Relay)

Link-Layer Address
: The variable length link-layer address. The Link-Layer Address is always specified in Canonical order.

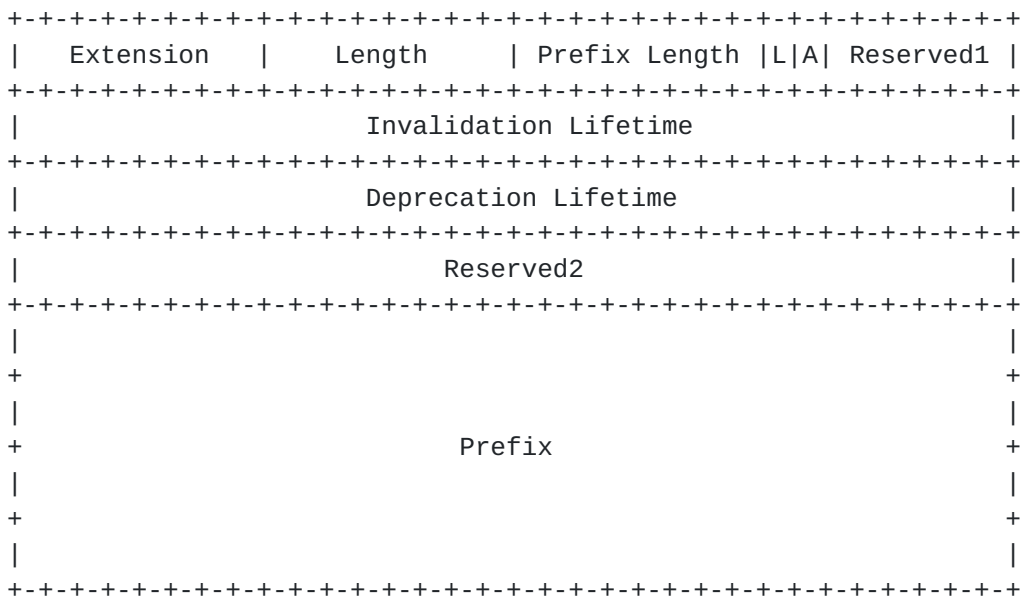
  The content of this field beyond the length specified by the address length field is unspecified and MUST be ignored by the receiver.

Description

The Source Link-Layer address extension contains the link-layer address of the sender of the packet. It is used in the Neighbor Solicitation, Router Solicitation, and Router Advertisement packets.

The Target Link-Layer address extension contains the link-layer address of the target. It is used in in Neighbor Advertisement and Redirect packets.

8.2. Prefix Information



Fields:

Extension 3

Length 3

Prefix Length 8-bit unsigned integer. The number of leading bits in the Prefix that are valid. The value ranges from 0 to 128.

L 1-bit on-link flag. When set, indicates that this prefix can be used for on-link determination.

A 1-bit address-configuration flag. When set indicates that this prefix can used for automatic address configuration as specified in [ADDRCONF].

Reserved1 6-bit unused field. It MUST be initialized to zero by the sender and ignored by the receiver.

Invalidation Lifetime
32-bit unsigned integer. The lifetime of the prefix in seconds for the purpose of on-link determination. This lifetime is also used by [ADDRCONF].

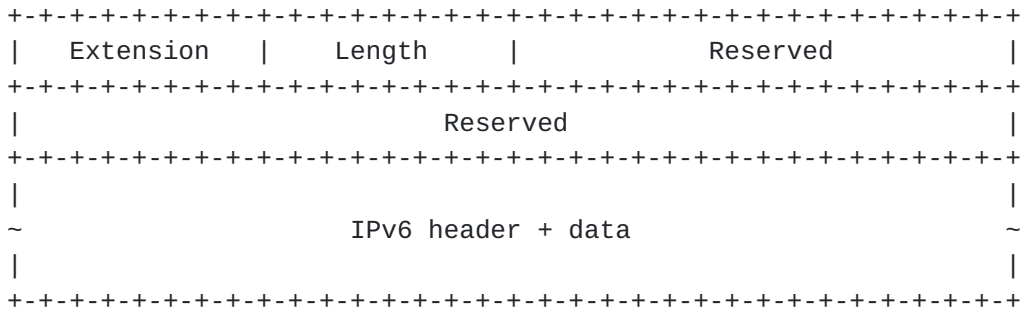
Deprecation Lifetime
32 bits reserved for automatic address configuration. See [ADDRCONF].

Reserved2 This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Prefix An IP address or a prefix of an IP address. The prefix length field contains the number of valid leading bits in the prefix.

Description
The Prefix Information extension is only used in Router Advertisement packets. It provide hosts with on-link prefixes and prefixes for Address Autoconfiguration.

8.3. Redirected Header



Fields:

Extension 4

Length The length of the extension in units of 8 octets.

Reserved These fields are unused. They MUST be initialized to zero by the sender and ignored by the receiver.

IPv6 header + data

The original packet truncated to ensure that the size of the redirect message does not exceed 576 octets.

Description

The Redirected Header extension MUST be included in Redirect packets.

8.4. Suggested Hop Limit

Fields:

Extension	5
Length	1
Hops	8-bit unsigned integer. The suggested hop limit.
Reserved	These fields are unused. They MUST be initialized to zero by the sender and ignored by the receiver.

Description

The Suggested Hop Limit extension MAY be included in Router Advertisement packets.

Hosts SHOULD handle this extension by computing the default Hop Limit as the maximum of all received Suggested Hop Limit extensions while ignoring those received from routers that have been timed out from the Default Router List.

8.5. Neighbor Unreachability Detection Timer

Fields:

Extension 6

Length 1

Timer 8-bit unsigned integer. The suggested Neighbor Unreachability Detection timer in seconds.

Reserved These fields are unused. They MUST be initialized to zero by the sender and ignored by the receiver.

Description

The Suggested Neighbor Unreachability Timer extension MAY be included in Router Advertisement packets.

Hosts SHOULD handle this extension by computing the REACHABLE_TIME as the minimum of all received Suggested Neighbor Unreachability Timers while ignoring those received from routers that have been timed out from the Default Router List.

If no Suggested Neighbor Unreachability Timer extension has been received (e.g. due to no routers on the link) the node MUST use the protocol constant defined in section 10.

8.6. MTU

Fields:

Extension 7

Length 1

MTU 16-bit unsigned integer. The recommended MTU for the link.

Reserved This field is unused. It MUST be initialized to zero by the sender and ignored by the receiver.

Description

The MTU extension SHOULD be included in Router Advertisement packets when the link has no well-known MTU and it MAY be included on links with a well-known MTU.

Hosts that operate on a link that does not have a well-defined MTU MUST handle this extension by computing the MTU of the link as the minimum of received MTU extensions while ignoring those received from routers that have been timed out from the Default Router List.

9. MULTIHOMED HOSTS

There are some special Neighbor Discovery rules and constraints that apply only to hosts that have multiple interfaces. Note that this section explicitly does not attempt to define the operation of multihomed hosts. It serves merely to point out some ND issues for multihomed hosts.

If a multihomed host hears no Router Advertisements at all (i.e. on none of its interfaces) the host can not determine which interface to use when sending packets. (A host with only one interface would assume that all destinations are on-link in this case.) Therefore multihomed hosts require that they can receive Router Advertisement on at least one of their interfaces. The exception to this is when the multihomed host is manually configured with the on-link prefixes for its interfaces.

If a multihomed host hears routers on a subset of its interfaces it will not send packets out any of the interfaces that do not have a router since it will not have received any prefixes for those links. Once again, the exception to this is when the multihomed host is manually configured with the on-link prefixes for the links that have no routers.

If a multihomed host hears no Prefix Information extensions from its routers it will not be able to make optimal interface selection when communicating with neighbors; without the prefixes the host can not tell which nodes are neighbors on which interfaces. It is recommended, and on multicast links required, that routers always advertise the on-link prefixes for the benefit of multihomed hosts.

10. PROTOCOL CONSTANTS

Router constants:

MAX_INITIAL_RTR_ADVERT_INTERVAL	16	seconds
MAX_INITIAL_RTR_ADVERTISEMENTS	3	transmissions
MAX_RTR_RESPONSE_DELAY	2	seconds

Host constants:

MAX_RTR_SOLICITATION_DELAY	1	second
RTR_SOLICITATION_INTERVAL	3	seconds
MAX_RTR_SOLICITATIONS	3	transmissions

Node constants:

RESOLVE_RETRANS_TIMER	1	second
UNREACHABLE_THRESHOLD	10	transmissions
MAX_NEIGHBOR_ADVERTISEMENTS	3	transmissions
MIN_NEIGHBOR_ADVERT_INTERVAL	16	seconds
REACHABLE_TIME	30	seconds
REACHABLE_RETRANS_TIME	1	second
DEFAULT_RTR_PROBE_INTERVAL	4	seconds
DELAY_FIRST_PROBE_TIME	4	seconds
CONSECUTIVE_UNICAST_PROBES	4	transmissions

Additional protocol constants are defined with the message formats in Section 5.1, 6.1, and 7.1.

All protocol constants are subject to change in future revisions of the protocol.

11. SECURITY CONSIDERATIONS

The Neighbor Discovery protocol packet exchanges can be authenticated

using the IPv6 Authentication Header [IPv6-AUTH].

It MUST be possible for the system administrator to configure a node to ignore any Neighbor Discovery messages that are not authenticated using either the Authentication Header or Encapsulating Security Payload. The configuration technique for this MUST be documented.

The trust model for redirects is based only trusting a redirect received from the current first hop node. It is natural to trust the routers on the link. If a host has been redirected to another host (i.e. the destination is on-link) there is no way to prevent the target from issuing another redirect to some other destination. However, this exposure is no worse than it was; the target host, once subverted, could always act as a hidden router to forward traffic elsewhere.

Confidentiality issues are addressed by the IP Security Architecture and the IP Encapsulating Security Payload documents [IPv6-SA, IPv6-ESP].

REFERENCES

[ADDRCONF] S. Thomson, "IPv6 Address Autoconfiguration", Internet Draft.

[ADDR-ARCH] S. Deering, R. Hinden, Editors, "IP Version 6 Addressing Architecture", Internet Draft.

[ANYCST] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.

[ARP] D. Plummer, "An Ethernet Address Resolution Protocol", STD 37, RFC 826, November 1982.

[HR-CL] R. Braden, Editor, "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[ICMPv4] J. Postel, "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[ICMPv6] A. Conta, and S. Deering, "ICMP for the Internet Protocol Version 6 (IPv6)", Internet Draft.

[IPv6] S. Deering, R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", Internet Draft.

[IPv6-SA] R. Atkinson. IPv6 Security Architecture. Internet Draft, March 1995.

[IPv6-AUTH] R. Atkinson. IPv6 Authentication Header. Internet Draft, March 1995.

[IPv6-ESP] R. Atkinson. IPv6 Encapsulating Security Payload. Internet Draft, February 1995.

[RDISC] S. Deering, "ICMP Router Discovery Messages", RFC 1256, September 1991.

[SH-MEDIA] R. Braden, J. Postel, Y. Rekhter, "Internet Architecture Extensions for Shared Media", RFC 1620, May 1994.

AUTHORS' ADDRESSES

Erik Nordmark
Sun Microsystems, Inc.
2550 Garcia Ave
Mt. View, CA 94041
USA

phone: +1 415 336 2788
fax: +1 415 336 6015
email: nordmark@sun.com

Thomas Narten
IBM Corporation
P.O. Box 12195
Research Triangle Park, NC 27709-2195
USA

phone: +1 919 254 7798
fax: +1 919 254 4027
email: narten@vnet.ibm.com

William Allen Simpson
Daydreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071
USA

email: Bill.Simpson@um.cc.umich.edu
bsimpson@MorningStar.com

CHANGES SINCE PREVIOUS DOCUMENT

This version of the "IPv6 Neighbor Discovery" includes several changes from the previous version documented in:

<draft-simpson-ipv6-discov-formats-02.txt>, and
<draft-simpson-ipv6-discov-process-02.txt>

The changes agreed to at working group meetings at Xerox Parc and at Danvers IETF:

- Renamed the Media-Access extension to be the Link-Layer Address extension.

- Use of different extensions for addresses that refer to the sender of the packet and the receiver instead of using the Known-Identifier extension for both.

- Changed the processing of General/Neighbor Solicitation in order to achieve 2 packet exchange just like ARP.

- Removed the Node-Heard extension.

Other changes:

- Merged the processing and format documents into a single document with an extensive introduction to the protocol.

- Aligned the document with [ADDRCONF]. In particular this implied the removal of the Change-Identifier extension.

- Off-link prefixes are not advertized in Router Advertisements (no simple routing protocol). This removes the need for a preference in the Prefix Information extension.

- Specified a more detailed Neighbor Unreachability Detection algorithm (used to be called Dead Node Detection).

- Removed the lifetime field from Neighbor Advertisements. The protocol uses Neighbor Unreachability Detection to time out state created by Neighbor Advertisements.

- Removed the Maximum Receive Unit fields from packets since per-node MTU (or MRU) links do not work with multicast. Instead routers send an MTU extension in order to handle links that do not have a well-defined MTU.

- Changed alignment mechanisms for extensions. All extensions

are a multiple of 8 octets. Thus there is no longer a need for pad extensions.

- Added support for anycast addresses.

- Removed the ability to redirect prefixes to simplify host processing.

- Removed lingering mobility support (Mobility-Support extension and Remote Redirect message.)

- All messages have separate ICMP types. Redirect type is now in the error range (<128) and the others in the information range (>=128)

- Moved fixed-length fields that are always present in a particular type of packet into the fixed header.

- Renamed "General" Solicitation/Advertisement to "Neighbor" Solicitation/Advertisement.

- Changed the default Router Advertisement period from 30 seconds to 600 seconds; same value as in RFC-1256. This change is possible since Neighbor Unreachability Detection will detect unreachable routers and switch a reachable router independent of the frequency of the Router Advertisements.

- Specified rules for when a node should generate ICMP address unreachable errors due to Address Resolution failures.

OPEN ISSUES

Misc issues:

- The protocol currently recommends (SHOULD) that nodes generate ICMP Address Unreachable errors when Address Resolution fails. The protocol requires that nodes retransmit Neighbor Solicitations in order to be able to generate such ICMP errors. ARP does not require retransmission of ARP requests.

  Is the utility of such errors high enough to warrant the use of a retransmission timer? Tools like 'ping' would report "Address Unreachable" errors instead of no response and end users would possibly see "Address Unreachable" errors rather than "timed out". In some cases applications might be able to try alternate addresses more quickly during connection opens. The latter may become more important as addresses come and go more quickly.

(Designated) addresses:

- ND requires that routers know the designed address for all other routers attached to the same link. Is this a reasonable requirement? What mechanism can routers use to learn this designated address from their peers? (routing protocols?, receiving Router Advertisements?)

- Should we require that the source addresses of all Neighbor Discovery packets be link-local? Link-local source addresses provides an extra level of robustness by preventing off-link nodes from generating bogus ND packets (assuming that routers don't forward packets with a link-local source address). This is more of an issue in v6 than v4 because v6 depends on ND messages to decide which destinations are on-link. Such a requirement would assume that link-local addresses exist on all types of links.

- Can we assume that a booting node will always be able construct a link-local address before it sends out a Router Solicitation packet? Routers ignore Router Solicitations from the unspecified IP address.

- Should we change the solicited-node multicast address range from FF02::0700-FF02::07FF to FF02::0100-FF02::01FF? Why was "07" selected?

Support for redundant (replicated) interfaces:

- Nodes can have redundant interfaces on the same link; how quickly does a neighbor have to be able to switch from using the link-layer

address of a faulty interface to another interface? The current specification has a separate mechanism that is only used to speed up this case: The use of multicast Neighbor Advertisements.

Can we just make NUD aggressive enough to detect the link layer address change, and remove the extra mechanism?

The Neighbor Unreachability Detection as currently specified will detect the link-layer address change but the switch over time is probably on the order of 1 minute.

NUD issues:

- Is the Neighbor Unreachability Detection algorithm simple enough? Is the description understandable?

- Currently only routers use designated addresses as source. If hosts have multiple addresses the NUD algorithm will treat each address as a separate neighbor, potentially causing redundant NUD probes.

  Should NA messages we changed to list all of a node's addresses so that hosts can keep track of the "equivalence class" of addresses that correspond to a single neighbor?

- How long should we retain the link-layer address after consecutive probes go unanswered? Should we keep the address when going to the TRY_ALTERNATES status in order to continue sending packets to the link-layer address even though explicit probes are not generating the desired reachability confirmation?

- What are the good values for various thresholds and timers that are used by NUD? Do some of these values have to be dynamic and/or settable by parameters in Router Advertisements in order to handle links with widely varying bandwidth and propagation delay?

ND support for mobility:

- What base level support does the (yet to be defined) mobility scheme require from ND, if any? In particular, what support is needed to handle mobiles that move off-link?

  This specification suggests using proxy Neighbor Advertisements for mobile nodes that move off-link since the proxy mechanism is very simple to implement in the hosts and it is already needed to support anycast addresses.

There is concern that proxy responses are hard to trust i.e. to make secure using authentication. An alternate model, which appears to be more complex, is to require that hosts switch to sending packets to a default router after Address Resolution and Neighbor Unreachability Detection fails.

There are two key differences between using a multicast Neighbor Solicitation with a proxy response and just sending the packet to one of the default routers:

- The latter requires that the routers maintain a well-synchronized distributed database since any router might receive a packet for any mobile. In the former scheme is it possible to partition the database; each router can support a subset of the mobiles and respond to solicitations for those nodes.
- The latter requires that all default routers participate in the mobility handling i.e. the distributed database. Even though we want all routers to be capable of acting as "home agents" an administrator might only enable this in a subset of the routers on the link. One reason for using a subset is that it presumably would reduce the database synchronization traffic.

If all default routers on a link MUST participate in the mobility support you don't have to add any complexity to the hosts. However, this might not be a realistic assumption.

Without this assumption, if you want the host to send to a default router after a NUD failure for an on-link destination the host has to be able to somehow handle default routers that are not in sync with the mobility database. This means that the host probably has to ignore (for some time after the NUD failure) a redirect that tells it that the destination is on-link and instead try a different default router.

An alternative would be for hosts to know which default routers are "mobility aware" and only used those routers after a NUD failure.

- How quickly does a node have to detect that a mobile neighbor has moved off-link? Can we just use NUD, as the protocol currently does, to detect this or do we need faster mechanisms? The Neighbor Unreachability Detection will detect the link-layer address change but the switch over time is probably on the order of 1 minute.

  Does mobility require a ND mechanism for mobile nodes to send a message that in effect says "I'm leaving the link, use the following agent instead"?

  Should the protocol allow multicast Neighbor Advertisement as an unreliable way of updating neighbors when a mobile has moved off-

link? This adds complexity and traffic especially when nodes have multiple IP addresses per interface.

NBMA/ATM link support:

- What additional support or additional text, if any, should the document contain about address resolution servers? (as used on ATM and SMDS; RFC-1577)

  One solution that is supported by ND as specified is to

  1) manually configure a Default Router List (which includes configuring the link-layer addresses of the routers), and

  2) not configure any on-link prefixes.

  This will make hosts send to the default routers and get redirected. The manually configured default routers could be the AR servers (which would redirect to the "real" routers), or every router could contain AR server functionality.

  The protocol as currently specified does not support the "inverse ARP" functionality in RFC 1577, which is used for

  - AR servers determining the IP addresses of the hosts, and

  - determining the peer's IP addresses on PVCs.

  A protocol extension could presumably be made either to have hosts periodically unicast NA to each default router on such networks, or allow unicast NS for the unspecified address. Do we want to address this issue? Should it be addressed in this document or can it be handled in a future document?

- How should ND specify ATM link-layer addresses that consist of an E.164 address plus an NSAP address? This is one of the address formats supported by RFC-1577. Is this form of address likely to ever be used?

Packet format issues:

- The extensions for MTU, NUD timer and hop-limit are not very space efficient. Should they be merged into a single extension? Should they be placed in the fixed part of the Router Advertisement packet? Both changes assume that we define a designated value for "unspecified" (e.g. 0) when the routers have nothing to say.

- The protocol specifies that Code = 0/1 is used in Neighbor Solicitation, Neighbor Advertisement, and Redirect messages to allow the receiver of the packet determine if the source (or redirection target in the case of a redirect) is a router or a host. This is strictly speaking only necessary in the Neighbor Advertisement message. The information is used by Neighbor Unreachability Detection to detect when a router has been converted to a host. Should the Code 0/1 distinction only be used in Neighbor Advertisements?

- How should link-layer addresses be encoded in the link-layer address extensions, in particular the addresses that consist of a string of decimal digits? (Such as E.164 addresses.) The current specification states that the family value implicitly defines whether the address length is in the unit of nibbles or bytes. Alternatives are:

  - Have an explicit flag that specifies "length is in nibbles" vs. "length is in bytes".

  - Always use nibbles as the unit (e.g. an IEEE 802 address would be 12 nibbles long).

  - Require that the decimal digit strings be encoded as one digit per byte (instead of BCD encoding) to force everything to be in units of bytes.

Other protocol processing issues:

- San Jose IETF resulted in millisecond granularity for lifetimes in order to match SNMP timer values. The February WG meeting at Xerox Parc resulted in extending them from 16 to 32 bits. What do we want to do?

  The current proposal has a 32-bit invalidation lifetime in seconds for prefixes and a 32-bit deprecation lifetime which is only used by [ADDRCONF].

- Power failure scenario: Should the protocol require that routers multicast delayed Router Advertisements in response to Router Solicitations in order to reduce the number of Router Advertisements when all hosts boot during a short time interval? The current specification says "MAY".

- Changes in advertised prefixes: Routers might want to send out immediate advertisements when the set of advertised prefixes changes.

Should the protocol allow this and, if so, what are the time constraints? (how frequently can this be done, etc)

Security issues:

- Proxy Neighbor Advertisements do not fit the trust model. Even if they are authenticated it is not possible for a host to determine if the router has authority to proxy for the target. We might be able to fix this by requiring that only routers (on the Default Router List) be allowed to send proxy responses.

- What is the trust model for anycast addresses i.e. how does a node know that a neighbor can claim to offer the anycast service?

- Should the authentication requirements be higher for Redirect messages than for other ND messages? Redirects can easily be used for denial of service attacks.

- Should ND somehow prefer authenticated packets over non-authenticated packets? (e.g. Neighbor Advertisements for the same target)

- What is the trust model for Router Advertisements i.e. in the presence of authentication how does a host now which neighbors are authorized to send Router Advertisements?