



# Windows Server® 2008

## Introduction to IP Version 6

*Microsoft Corporation*

*Published: September 2003*

*Updated: January 2008*

---

### **Abstract**

Due to recent concerns over the impending depletion of the current pool of Internet addresses and the desire to provide additional functionality for modern devices, an upgrade of the current version of the Internet Protocol (IP), called IPv4, has been defined. This new version, called IP version 6 (IPv6), resolves unanticipated IPv4 design issues and takes the Internet into the 21<sup>st</sup> Century. This paper describes the problems of the IPv4 Internet and how they are solved by IPv6, IPv6 addressing, the new IPv6 header and its extensions, the IPv6 replacements for the Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP), neighboring node interaction, and IPv6 address autoconfiguration. This paper provides a foundation of Internet standards-based IPv6 concepts and is intended for network engineers and support professionals who are already familiar with basic networking concepts and TCP/IP.

**Microsoft**

*The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.*

*Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.*

*© 2008 Microsoft Corporation. All rights reserved.*

*Microsoft, Windows, Windows Server, Windows Vista, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

---

## Contents

<b>Introduction</b> .....	<b>1</b>
IPv6 Features.....	2
New Header Format.....	2
Large Address Space.....	2
Efficient and Hierarchical Addressing and Routing Infrastructure.....	2
Stateless and Stateful Address Configuration.....	3
Built-in Security.....	3
Better Support for Prioritized Delivery.....	3
New Protocol for Neighboring Node Interaction.....	3
Extensibility.....	3
Differences Between IPv4 and IPv6.....	3
IPv6 Packets over LAN Media.....	4
Ethernet II Encapsulation.....	5
IEEE 802.3, IEEE 802.5, and FDDI Encapsulation.....	5
IPv6 Implementations from Microsoft.....	6
The Next Generation TCP/IP Stack in Windows Vista and Windows Server 2008.....	6
The IPv6 Protocol for the Windows Server 2003 Family, Windows XP, and Windows CE .NET.....	7
Non-production IPv6 Implementations from Microsoft.....	7
<b>IPv6 Addressing</b> .....	<b>8</b>
The IPv6 Address Space.....	8
IPv6 Address Syntax.....	8
Compressing Zeros.....	9
IPv6 Prefixes.....	9
Types of IPv6 Addresses.....	9
Links and Subnets.....	10
Unicast IPv6 Addresses.....	10
Global Unicast Addresses.....	10
Local-Use Unicast Addresses.....	11
Zone IDs for Local-Use Addresses.....	13
Unique Local IPv6 Unicast Addresses.....	14

Special IPv6 Addresses .....	15
Compatibility Addresses .....	15
Multicast IPv6 Addresses .....	16
Solicited-Node Address .....	17
Anycast IPv6 Addresses .....	18
IPv6 Addresses for a Host .....	18
IPv6 Addresses for a Router .....	19
IPv6 Interface Identifiers .....	19
EUI-64 address-based interface identifiers .....	20
Temporary Address Interface Identifiers .....	23
Mapping IPv6 Multicast Addresses to Ethernet Addresses .....	23
IPv6 and DNS .....	24
The Host Address (AAAA) Resource Record .....	24
The IP6.ARPA Domain .....	25
Source and Destination Address Selection .....	25
IPv4 Addresses and IPv6 Equivalents .....	25
<b>IPv6 Header .....</b>	<b>27</b>
Structure of an IPv6 Packet .....	27
IPv6 Header .....	27
Extension Headers .....	27
Upper Layer Protocol Data Unit .....	27
IPv4 Header .....	27
IPv6 Header .....	29
Values of the Next Header Field .....	30
Comparing the IPv4 and IPv6 Headers .....	31
IPv6 Extension Headers .....	31
Extension Headers Order .....	32
Hop-by-Hop Options Header .....	33
Destination Options Header .....	33
Routing Header .....	34
Fragment Header .....	35
Authentication Header .....	36

Encapsulating Security Payload Header and Trailer .....	37
IPv6 MTU .....	37
Upper Layer Checksums .....	38
<b>ICMPv6.....</b>	<b>39</b>
Types of ICMPv6 Messages .....	39
ICMPv6 Header.....	39
ICMPv6 Error Messages .....	40
Destination Unreachable .....	40
Packet Too Big .....	41
Time Exceeded .....	42
Parameter Problem .....	42
ICMPv6 Informational Messages .....	43
Echo Request .....	43
Echo Reply .....	43
Comparing ICMPv4 and ICMPv6 Error Messages .....	44
Path MTU Discovery .....	45
Changes in Path MTU .....	45
<b>Multicast Listener Discovery.....</b>	<b>46</b>
MLD Messages .....	46
Multicast Listener Query.....	46
Multicast Listener Report.....	47
Multicast Listener Done.....	48
MLDv2.....	48
<b>Neighbor Discovery.....</b>	<b>49</b>
Neighbor Discovery Message Format .....	50
Neighbor Discovery Options .....	51
Source/Target Link-Layer Address Option.....	51
Prefix Information Option.....	52
Redirected Header Option.....	53
MTU Option .....	54
Neighbor Discovery Messages .....	55
Router Solicitation .....	55

Router Advertisement.....	56
Neighbor Solicitation .....	58
Neighbor Advertisement.....	59
Redirect .....	61
Neighbor Discovery Processes.....	62
Address Resolution .....	63
Duplicate Address Detection .....	64
Router Discovery.....	66
Neighbor Unreachability Detection.....	68
Redirect Function .....	70
Host Sending Algorithm .....	72
<b>Address Autoconfiguration .....</b>	<b>74</b>
Autoconfigured Address States .....	74
Types of Autoconfiguration .....	75
Autoconfiguration Process .....	75
DHCPv6 .....	78
DHCPv6 Messages .....	79
DHCPv6 Support in Windows.....	79
<b>IPv6 Routing.....</b>	<b>80</b>
Contents of an IPv6 Routing Table.....	80
Route Determination Process.....	81
Example IPv6 Routing Tables for Windows Vista and Windows Server 2008 .....	82
The Route Print Command.....	82
The netsh interface ipv6 show route Command.....	83
<b>Summary .....</b>	<b>85</b>
<b>Related Links .....</b>	<b>86</b>

---

## Introduction

The current version of IP (known as Version 4 or IPv4) has not been substantially changed since RFC 791 was published in 1981. IPv4 has proven to be robust, easily implemented and interoperable, and has stood the test of scaling an internetwork to a global utility the size of today's Internet. This is a tribute to its initial design.

However, the initial design did not anticipate the following:

- The recent exponential growth of the Internet and the impending exhaustion of the IPv4 address space.

IPv4 addresses have become relatively scarce, forcing some organizations to use a Network Address Translator (NAT) to map multiple private addresses to a single public IP address. While NATs promote reuse of the private address space, they do not support standards-based network layer security or the correct mapping of all higher layer protocols and can create problems when connecting two organizations that use the private address space.

Additionally, the rising prominence of Internet-connected devices and appliances ensures that the public IPv4 address space will eventually be depleted.

- The growth of the Internet and the ability of Internet backbone routers to maintain large routing tables.

Because of the way that IPv4 address prefixes have been and are currently allocated, there are routinely over 85,000 routes in the routing tables of Internet backbone routers. The current IPv4 Internet routing infrastructure is a combination of both flat and hierarchical routing.

- The need for simpler configuration.

Most current IPv4 implementations must be either manually configured or use a stateful address configuration protocol such as Dynamic Host Configuration Protocol (DHCP). With more computers and devices using IP, there is a need for a simpler and more automatic configuration of addresses and other configuration settings that do not rely on the administration of a DHCP infrastructure.

- The requirement for security at the IP level.

Private communication over a public medium like the Internet requires encryption services that protect the data being sent from being viewed or modified in transit. Although a standard now exists for providing security for IPv4 packets (known as Internet Protocol security or IPsec), this standard is optional and proprietary solutions are prevalent.

- The need for better support for real-time delivery of data—also called quality of service (QoS).

While standards for QoS exist for IPv4, real-time traffic support relies on the IPv4 Type of Service (TOS) field and the identification of the payload, typically using a UDP or TCP port. Unfortunately, the IPv4 TOS field has limited functionality and over time there were various local interpretations. In addition, payload identification using a TCP and UDP port is not possible when the IPv4 packet payload is encrypted.

To address these and other concerns, the Internet Engineering Task Force (IETF) has developed a suite of protocols and standards known as IP version 6 (IPv6). This new version, previously called IP-The Next Generation (IPng), incorporates the concepts of many proposed methods for updating the

IPv4 protocol. The design of IPv6 is intentionally targeted for minimal impact on upper and lower layer protocols by avoiding the random addition of new features.

## IPv6 Features

The following are the features of the IPv6 protocol:

- New header format
- Large address space
- Efficient and hierarchical addressing and routing infrastructure
- Stateless and stateful address configuration
- Built-in security
- Better support for prioritized delivery
- New protocol for neighboring node interaction
- Extensibility

The following sections discuss each of these new features in detail.

### New Header Format

The IPv6 header has a new format that is designed to keep header overhead to a minimum. This is achieved by moving both non-essential fields and optional fields to extension headers that are placed after the IPv6 header. The streamlined IPv6 header is more efficiently processed at intermediate routers.

IPv4 headers and IPv6 headers are not interoperable. IPv6 is not a superset of functionality that is backward compatible with IPv4. A host or router must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats. The new IPv6 header is only twice as large as the IPv4 header, even though IPv6 addresses are four times as large as IPv4 addresses.

### Large Address Space

IPv6 has 128-bit (16-byte) source and destination IP addresses. Although 128 bits can express over  $3.4 \times 10^{38}$  possible combinations, the large address space of IPv6 has been designed to allow for multiple levels of subnetting and address allocation from the Internet backbone to the individual subnets within an organization.

Even though only a small number of the possible addresses are currently allocated for use by hosts, there are plenty of addresses available for future use. With a much larger number of available addresses, address-conservation techniques, such as the deployment of NATs, are no longer necessary.

### Efficient and Hierarchical Addressing and Routing Infrastructure

IPv6 global addresses used on the IPv6 portion of the Internet are designed to create an efficient, hierarchical, and summarizable routing infrastructure that is based on the common occurrence of multiple levels of Internet service providers.

## Stateless and Stateful Address Configuration

To simplify host configuration, IPv6 supports both stateful address configuration, such as address configuration in the presence of a DHCP server, and stateless address configuration (address configuration in the absence of a DHCP server). With stateless address configuration, hosts on a link automatically configure themselves with IPv6 addresses for the link (called link-local addresses) and with addresses derived from prefixes advertised by local routers. Even in the absence of a router, hosts on the same link can automatically configure themselves with link-local addresses and communicate without manual configuration.

## Built-in Security

Support for IPsec is an IPv6 protocol suite requirement. This requirement provides a standards-based solution for network security needs and promotes interoperability between different IPv6 implementations.

## Better Support for Prioritized Delivery

New fields in the IPv6 header define how traffic is handled and identified. Traffic identification using a Flow Label field in the IPv6 header allows routers to identify and provide special handling for packets belonging to a flow, a series of packets between a source and destination. Because the traffic is identified in the IPv6 header, support for prioritized delivery can be achieved even when the packet payload is encrypted with IPsec.

## New Protocol for Neighboring Node Interaction

The Neighbor Discovery protocol for IPv6 is a series of Internet Control Message Protocol for IPv6 (ICMPv6) messages that manage the interaction of neighboring nodes (nodes on the same link). Neighbor Discovery replaces the broadcast-based Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and ICMPv4 Redirect messages with efficient multicast and unicast Neighbor Discovery messages.

## Extensibility

IPv6 can easily be extended for new features by adding extension headers after the IPv6 header. Unlike options in the IPv4 header, which can only support 40 bytes of options, the size of IPv6 extension headers is only constrained by the size of the IPv6 packet.

## Differences Between IPv4 and IPv6

Table 1 highlights some of the key differences between IPv4 and IPv6.

**Table 1 Differences between IPv4 and IPv6**

IPv4	IPv6
Source and destination addresses are 32 bits (4 bytes) in length.	Source and destination addresses are 128 bits (16 bytes) in length. For more information, see "IPv6 Addressing."
IPsec support is optional.	IPsec support is required. For more information, see "IPv6 Header."
No identification of packet flow for QoS handling	Packet flow identification for QoS handling by

by routers is present within the IPv4 header.	routers is included in the IPv6 header using the Flow Label field. For more information, see "IPv6 Header."
Fragmentation is done by both routers and the sending host.	Fragmentation is not done by routers, only by the sending host. For more information, see "IPv6 Header."
Header includes a checksum.	Header does not include a checksum. For more information, see "IPv6 Header."
Header includes options.	All optional data is moved to IPv6 extension headers. For more information, see "IPv6 Header."
Address Resolution Protocol (ARP) uses broadcast ARP Request frames to resolve an IPv4 address to a link layer address.	ARP Request frames are replaced with multicast Neighbor Solicitation messages. For more information, see "Neighbor Discovery."
Internet Group Management Protocol (IGMP) is used to manage local subnet group membership.	IGMP is replaced with Multicast Listener Discovery (MLD) messages. For more information, see "Multicast Listener Discovery."
ICMP Router Discovery is used to determine the IPv4 address of the best default gateway and is optional.	ICMP Router Discovery is replaced with ICMPv6 Router Solicitation and Router Advertisement messages and is required. For more information, see "Neighbor Discovery."
Broadcast addresses are used to send traffic to all nodes on a subnet.	There are no IPv6 broadcast addresses. Instead, a link-local scope all-nodes multicast address is used. For more information, see "Multicast IPv6 Addresses."
Must be configured either manually or through DHCP.	Does not require manual configuration or DHCP. For more information, see "Address Autoconfiguration."
Uses host address (A) resource records in the Domain Name System (DNS) to map host names to IPv4 addresses.	Uses host address (AAAA) resource records in the Domain Name System (DNS) to map host names to IPv6 addresses. For more information, see "IPv6 and DNS."
Uses pointer (PTR) resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names.	Uses pointer (PTR) resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names. For more information, see "IPv6 and DNS."
Must support a 576-byte packet size (possibly fragmented).	Must support a 1280-byte packet size (without fragmentation). For more information, see "IPv6 MTU."

## IPv6 Packets over LAN Media

A link layer frame containing an IPv6 packet consists of the following structure:

- Link Layer Header and Trailer – The encapsulation placed on the IPv6 packet at the link layer.
- IPv6 Header – The new IPv6 header. For more information, see "IPv6 Header."
- Payload –The payload of the IPv6 packet. For more information, see "IPv6 Header."

Figure 1 shows the structure of a link layer frame containing an IPv6 packet.

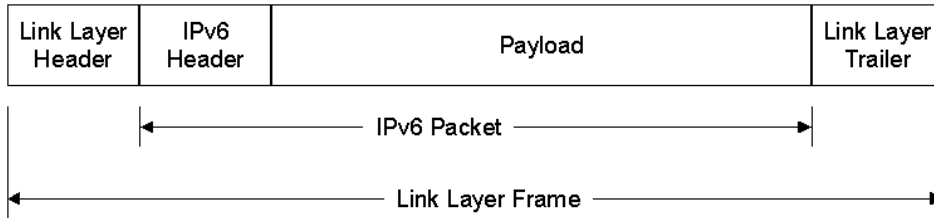


Figure 1 IPv6 packets at the link layer

For typical LAN technologies such as Ethernet, Token Ring, and Fiber Distributed Data Interface (FDDI), IPv6 packets are encapsulated in one of two ways—with either the Ethernet II header or a Sub-Network Access Protocol (SNAP) header used by IEEE 802.3 (Ethernet), IEEE 802.5 (Token Ring), and FDDI.

### Ethernet II Encapsulation

With Ethernet II encapsulation, IPv6 packets are indicated by setting the EtherType field in the Ethernet II header to 0x86DD (IPv4 is indicated by setting the EtherType field to 0x800). With Ethernet II encapsulation, IPv6 packets can have a minimum size of 46 bytes and a maximum size of 1,500 bytes. Figure 2 shows Ethernet II encapsulation for IPv6 packets.

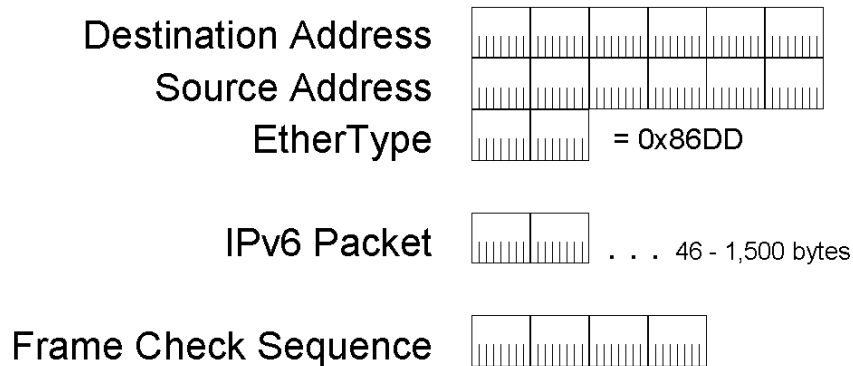


Figure 2 Ethernet II encapsulation

### IEEE 802.3, IEEE 802.5, and FDDI Encapsulation

On IEEE 802.3 (Ethernet), IEEE 802.5 (Token Ring), and FDDI networks, the Sub-Network Access Protocol (SNAP) header is used and the EtherType field is set to 0x86DD to indicate IPv6. Figure 3 shows SNAP encapsulation.

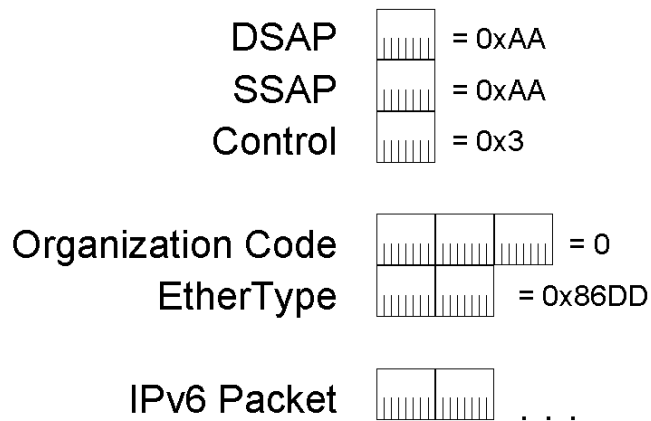


Figure 3 SNAP encapsulation used for IEEE 802.3, IEEE 802.5, and FDDI

For IEEE 802.3 encapsulation using the SNAP header, IPv6 packets can have a minimum size of 38 bytes and a maximum size of 1,492 bytes. For FDDI encapsulation using the SNAP header, IPv6 packets can have a maximum size of 4,352 bytes. For information on maximum IPv6 packet sizes for IEEE 802.5 links, see RFC 2470.

## IPv6 Implementations from Microsoft

Microsoft® has the following implementations of IPv6:

- The Next Generation TCP/IP stack in Windows Vista™ and Windows Server® 2008.
- The IPv6 protocol for the Windows Server 2003 family.
- The IPv6 protocol for Windows® XP Service Pack 1 (SP1) and later.
- The IPv6 protocol for Windows CE .NET version 4.1 and later.

The capture and parsing of IPv6 traffic is supported by Microsoft Network Monitor.

For all the IPv6 implementations from Microsoft, you can use IPv6 without affecting IPv4 communications.

## The Next Generation TCP/IP Stack in Windows Vista and Windows Server 2008

Microsoft Windows Vista and Windows Server 2008 include a new implementation of the TCP/IP protocol suite known as the Next Generation TCP/IP stack. The implementations of IPv6 prior to Windows Vista and Windows Server 2008 use a dual IPv4 and IPv6 stack architecture. For IPv6 support, you have to install a separate protocol through the Network Connections folder. The separate IPv6 protocol stack had its own Transport layer that included Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) and its own framing layer, which performs link layer encapsulation and decapsulation. The Next Generation TCP/IP stack supports the dual IP layer architecture in which the IPv4 and IPv6 implementations share common Transport and framing layers. The Next Generation TCP/IP stack has both IPv4 and IPv6 installed and enabled by default. There is no need to install a separate component to obtain IPv6 support.

For more information, see [Next Generation TCP/IP Stack in Windows Vista and Windows Server 2008](http://www.microsoft.com/technet/community/columns/cableguy/cg0905.mspx)" at <http://www.microsoft.com/technet/community/columns/cableguy/cg0905.mspx> and [Changes to IPv6](#)

in [Windows Vista and Windows Server 2008](#) at

<http://www.microsoft.com/technet/community/columns/cableguy/cg1005.mspx>.

### **The IPv6 Protocol for the Windows Server 2003 Family, Windows XP, and Windows CE .NET**

The IPv6 protocol for the Windows Server 2003 Family, Windows XP with SP1 and later, and Windows CE .NET is a production-quality implementation capable of supporting a set of key scenarios and can be installed and uninstalled as a network protocol through the Network Connections folder.

The IPv6 protocol for the Windows Server 2003 Family, Windows XP with SP1 and later, and Windows CE .NET is supported by Microsoft PSS for production use. For more information about applications and components that are IPv6-capable see Help on each of these platforms.

### **Non-production IPv6 Implementations from Microsoft**

The following IPv6 implementations have been released for non-production purposes only:

- The IPv6 Protocol for Windows XP with no service packs installed.
- The Microsoft IPv6 Technology Preview for Windows 2000 Service Pack 1 and later, available at <http://msdn.microsoft.com/downloads/sdks/platform/tpipv6.asp>.
- The Microsoft Research IPv6 Implementation, available at <http://www.research.microsoft.com/msripv6/>.

These implementations of IPv6 are not supported for production use by Microsoft Support Services. Check the individual Web sites and Help for information about reporting bugs and sending feedback to the Microsoft product group.

**Note** Microsoft has no plans to provide IPv6 implementations for Windows 98 or Windows Millennium Edition, or to provide a production-quality IPv6 implementation for Windows 2000.

---

## IPv6 Addressing

In this section, we examine:

- The IPv6 address space
- IPv6 address syntax
- IPv6 prefixes
- Types of IPv6 addresses
- Unicast IPv6 addresses
- Multicast IPv6 addresses
- Anycast IPv6 addresses
- IPv6 addresses for a host
- IPv6 addresses for a router
- IPv6 interface identifiers

### The IPv6 Address Space

The most obvious distinguishing feature of IPv6 is its use of much larger addresses. The size of an address in IPv6 is 128 bits, which is four times the larger than an IPv4 address. A 32-bit address space allows for  $2^{32}$  or 4,294,967,296 possible addresses. A 128-bit address space allows for  $2^{128}$  or 340,282,366,920,938,463,374,607,431,768,211,456 (or  $3.4 \times 10^{38}$  or 340 undecillion) possible addresses.

In the late 1970s when the IPv4 address space was designed, it was unimaginable that it could be exhausted. However, due to changes in technology and an allocation practice that did not anticipate the recent explosion of hosts on the Internet, the IPv4 address space was consumed to the point that by 1992 it was clear a replacement would be necessary.

With IPv6, it is even harder to conceive that the IPv6 address space will be consumed. To help put this number in perspective, a 128-bit address space provides 655,570,793,348,866,943,898,599 ( $6.5 \times 10^{23}$ ) addresses for every square meter of the Earth's surface.

It is important to remember that the decision to make the IPv6 address 128 bits in length was not so that every square meter of the Earth could have  $6.5 \times 10^{23}$  addresses. Rather, the relatively large size of the IPv6 address is designed to be subdivided into hierarchical routing domains that reflect the topology of the modern-day Internet. The use of 128 bits allows for multiple levels of hierarchy and flexibility in designing hierarchical addressing and routing that is currently lacking on the IPv4-based Internet.

The IPv6 addressing architecture is described in RFC 4291.

### IPv6 Address Syntax

IPv4 addresses are represented in dotted-decimal format. This 32-bit address is divided along 8-bit boundaries. Each set of 8 bits is converted to its decimal equivalent and separated by periods. For IPv6, the 128-bit address is divided along 16-bit boundaries, and each 16-bit block is converted to a 4-

digit hexadecimal number and separated by colons. The resulting representation is called colon-hexadecimal.

The following is an IPv6 address in binary form:

```
001000000000000100001101101110000000000000000010111100111011
0000001010101010000000001111111111111110001010001001110001011010
```

The 128-bit address is divided along 16-bit boundaries:

```
0010000000000001 0000110110111000 0000000000000000 0010111100111011 0000001010101010
0000000011111111 1111111000101000 1001110001011010
```

Each 16-bit block is converted to hexadecimal and delimited with colons. The result is:

```
2001:0DB8:0000:2F3B:02AA:00FF:FE28:9C5A
```

IPv6 representation can be further simplified by removing the leading zeros within each 16-bit block. However, each block must have at least a single digit. With leading zero suppression, the address representation becomes:

```
2001:DB8:0:2F3B:2AA:FF:FE28:9C5A
```

### Compressing Zeros

Some types of addresses contain long sequences of zeros. To further simplify the representation of IPv6 addresses, a contiguous sequence of 16-bit blocks set to 0 in the colon hexadecimal format can be compressed to “::”, known as *double-colon*.

For example, the link-local address of FE80:0:0:0:2AA:FF:FE9A:4CA2 can be compressed to FE80::2AA:FF:FE9A:4CA2. The multicast address FF02:0:0:0:0:0:2 can be compressed to FF02::2.

Zero compression can only be used to compress a single contiguous series of 16-bit blocks expressed in colon hexadecimal notation. You cannot use zero compression to include part of a 16-bit block. For example, you cannot express FF02:30:0:0:0:0:5 as FF02:3::5. The correct representation is FF02:30::5.

To determine how many 0 bits are represented by the “::”, you can count the number of blocks in the compressed address, subtract this number from 8, and then multiply the result by 16. For example, in the address FF02::2, there are two blocks (the “FF02” block and the “2” block.) The number of bits expressed by the “::” is 96 ( $96 = (8 - 2) \times 16$ ).

Zero compression can only be used once in a given address. Otherwise, you could not determine the number of 0 bits represented by each instance of “::”.

### IPv6 Prefixes

The prefix is the part of the address that indicates the bits that have fixed values or are the bits of the subnet prefix. Prefixes for IPv6 subnets, routes, and address ranges are expressed in the same way as Classless Inter-Domain Routing (CIDR) notation for IPv4. An IPv6 prefix is written in *address/prefix-length* notation. For example, 21DA:D3::/48 and 21DA:D3:0:2F3B::/64 are IPv6 address prefixes.

**Note** IPv4 implementations commonly use a dotted decimal representation of the network prefix known as the subnet mask. A subnet mask is not used for IPv6. Only the prefix length notation is supported.

### Types of IPv6 Addresses

There are three types of IPv6 addresses:

### 1. Unicast

A unicast address identifies a single interface within the scope of the type of unicast address. With the appropriate unicast routing topology, packets addressed to a unicast address are delivered to a single interface.

### 2. Multicast

A multicast address identifies multiple interfaces. With the appropriate multicast routing topology, packets addressed to a multicast address are delivered to all interfaces that are identified by the address. A multicast address is used for one-to-many communication, with delivery to multiple interfaces.

### 3. Anycast

An anycast address identifies multiple interfaces. With the appropriate routing topology, packets addressed to an anycast address are delivered to a single interface, the nearest interface that is identified by the address. The “nearest” interface is defined as being closest in terms of routing distance. An anycast address is used for one-to-one-of-many communication, with delivery to a single interface.

In all cases, IPv6 addresses identify interfaces, not nodes. A node is identified by any unicast address assigned to one of its interfaces.

**Note** RFC 4291 does not define a broadcast address. All types of IPv4 broadcast addressing are performed in IPv6 using multicast addresses. For example, the subnet and limited broadcast addresses from IPv4 are replaced with the link-local scope all-nodes multicast address of FF02::1.

## Links and Subnets

Similar to IPv4, an IPv6 subnet prefix is assigned to a single link. Multiple subnet prefixes can be assigned to the same link. This technique is called *multinetting*.

## Unicast IPv6 Addresses

The following types of addresses are unicast IPv6 addresses:

- Global unicast addresses
- Link-local addresses
- Site-local addresses
- Unique local IPv6 unicast addresses
- Special addresses

### Global Unicast Addresses

Global unicast addresses are equivalent to public IPv4 addresses. They are globally routable and reachable on the IPv6 portion of the Internet. Unlike the current IPv4-based Internet, which is a mixture of both flat and hierarchical routing, the IPv6-based Internet has been designed from its foundation to support efficient, hierarchical addressing and routing. The scope, the portion of the IPv6 internetwork over which the address is unique, of a global unicast address is the entire IPv6 Internet.

Figure 4 shows the structure of global unicast addresses currently being allocated by IANA, as defined in RFC 3587.

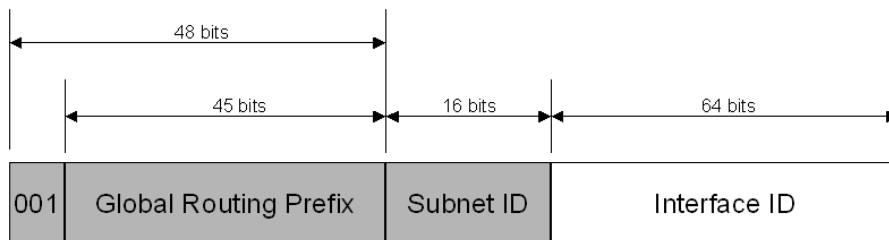


Figure 4 The global unicast address as defined in RFC 3587

The fields in the global unicast address are the following:

**Fixed portion set to 001** – The three high-order bits are set to 001. The address prefix for currently assigned global addresses is 2000::/3.

**Global Routing Prefix** – Indicates the global routing prefix for a specific organization's site. The combination of the three fixed bits and the 45-bit Global Routing Prefix is used to create a 48-bit site prefix, which is assigned to an individual site of an organization. Once assigned, routers on the IPv6 Internet forward IPv6 traffic matching the 48-bit prefix to the routers of the organization's site.

**Subnet ID** – The Subnet ID is used within an organization's site to identify subnets. The size of this field is 16 bits. The organization's site can use these 16 bits within its site to create 65,536 subnets or multiple levels of addressing hierarchy and an efficient routing infrastructure.

**Interface ID** – Indicates the interface on a specific subnet within the site. The size of this field is 64 bits.

The fields within the global unicast address create a three-level structure shown in Figure 5.

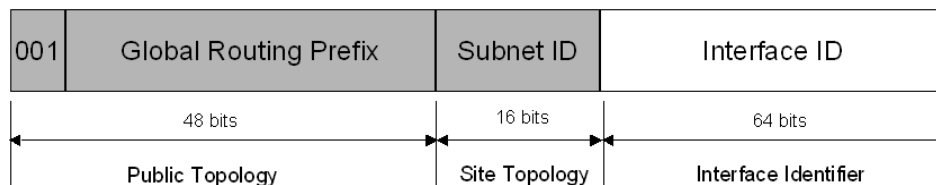


Figure 5 The three-level structure of the global unicast address

The public topology is the collection of larger and smaller ISPs that provide access to the IPv6 Internet. The site topology is the collection of subnets within an organization's site. The interface identifier identifies a specific interface on a subnet within an organization's site. For more information about global unicast addresses, see RFC 3587.

### Local-Use Unicast Addresses

There are two types of local-use unicast addresses:

1. Link-local addresses are used between on-link neighbors and for Neighbor Discovery processes.
2. Site-local addresses are used between nodes communicating with other nodes in the same site.

### Link-Local Addresses

Link-local addresses are used by nodes when communicating with neighboring nodes on the same link. For example, on a single link IPv6 network with no router, link-local addresses are used to communicate between hosts on the link. IPv6 link-local addresses are equivalent to IPv4 link-local addresses defined in RFC 3927 that use the 169.254.0.0/16 prefix. IPv4 link-local addresses are known as Automatic Private IP Addressing (APIPA) addresses for computers running current Microsoft Windows operating systems. The scope of a link-local address is the local link.

A link-local address is required for Neighbor Discovery processes and is always automatically configured, even in the absence of all other unicast addresses. For more information on the address autoconfiguration process for link-local addresses, see “Address Autoconfiguration.”

Figure 6 shows the structure of the link-local address.

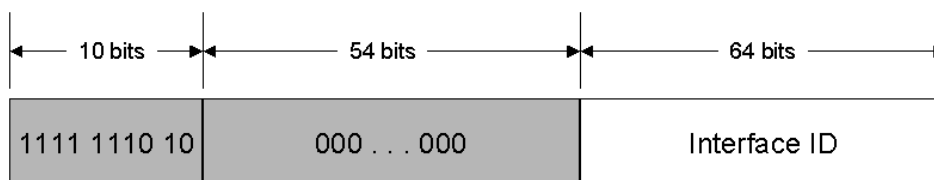


Figure 6 The link-local address

Link-local addresses always begin with FE80. With the 64-bit interface identifier, the prefix for link-local addresses is always FE80::/64. An IPv6 router never forwards link-local traffic beyond the link.

### Site-Local Addresses

Site-local addresses are equivalent to the IPv4 private address space (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16). For example, private intranets that do not have a direct, routed connection to the IPv6 Internet can use site-local addresses without conflicting with global unicast addresses. Site-local addresses are not reachable from other sites, and routers must not forward site-local traffic outside the site. Site-local addresses can be used in addition to global unicast addresses. The scope of a site-local address is the site. A site is an organization network or portion of an organization's network that has a defined geographical location (such as an office, an office complex, or a campus).

Unlike link-local addresses, site-local addresses are not automatically configured and must be assigned either through stateless or stateful address configuration processes. For more information, see “Address Autoconfiguration.”

Figure 7 shows the structure of the site-local address.

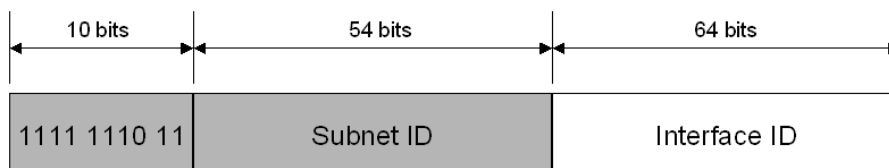


Figure 7 The site-local address

The first 10-bits are always fixed for site-local addresses (FEC0::/10). After the 10 fixed bits is a Subnet ID field that provides 54 bits with which you can create a hierarchical and summarizable routing

infrastructure within the site. After the Subnet ID field is a 64-bit Interface ID field that identifies a specific interface on a subnet.

---

**Note** RFC 3879 formally deprecates the use of site-local addresses for future IPv6 implementations. Existing implementations of IPv6 can continue to use site-local addresses.

---

### Zone IDs for Local-Use Addresses

Unlike global addresses, local-use addresses can be reused. Link-local addresses are reused on each link. Site-local addresses can be reused within each site of an organization. Because of this address reuse capability, link-local and site-local addresses are ambiguous. To specify which link on which an address is assigned or located or within which site an address is assigned or located, an additional identifier is needed. This additional identifier is a zone identifier (ID), also known as a scope ID, which identifies a connected portion of a network that has a specified scope. The syntax specified in RFC 4007 for identifying the zone associated with a local-use address is the following:

*Address%zone\_ID*

*Address* is a local-use address and *zone\_ID* is an integer value representing the zone. The values of the zone ID are defined relative to the sending host. Therefore, different hosts might determine different zone ID values for the same physical zone. For example, Host A might choose 3 to represent the zone ID of an attached link and Host B might choose 4 to represent the same link.

For Windows-based IPv6 hosts, the zone IDs for link-local and site-local addresses are defined as follows:

- For link-local addresses, the zone ID is typically the interface index of the interface either assigned the address or to be used as the sending interface for a link-local destination. The interface index is an integer starting at 1 that is assigned to IPv6 interfaces, which include a loopback and one or multiple tunnel or LAN interfaces. You can view the list of interface indexes from the display of the **netsh interface ipv6 show interface** command.
- For site-local addresses, the zone ID is the site ID, an integer assigned to the site of an organization. For organizations that do not reuse the site-local address prefix, the site ID is set to 1 by default and does not need to be specified. You can view the site ID from the display of the **netsh interface ipv6 show address level=verbose** command.

The following are examples of using Windows tools and the zone ID:

- **ping fe80::2b0:d0ff:fee9:4143%3** In this case, 3 is the interface index of the interface attached to the link containing the destination address.
- **tracert fec0::f282:2b0:d0ff:fee9:4143%2** In this case, 2 is the site ID of the organization site containing the destination address.

In Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008, the Ipconfig.exe tool displays the zone ID of local-use IPv6 addresses. The following is an excerpt from the display of the **ipconfig** command:

Ethernet adapter Local Area Connection:

```

Connection-specific DNS Suffix . : wcoast.example.com
IP Address. . . . . : 157.60.14.219
Subnet Mask . . . . . : 255.255.255.0

```

```

IP Address. . . . . : 3ffe:ffff:2a1c:2:1cc8:ef1d:1dd9:8066
IP Address. . . . . : 3ffe:ffff:2a1c:204:5aff:fe56:f5b
IP Address. . . . . : fe80::204:5aff:fe56:f5b%4
Default Gateway . . . . . : 157.60.14.1
                             fe80::20a:42ff:feb0:5400%4
    
```

For the link-local addresses in the display of the **ipconfig** command, the zone ID indicates the interface index of the interface either assigned the address (for IP Address) or the interface through which an address is reachable (for Default Gateway).

### Unique Local IPv6 Unicast Addresses

Site-local addresses provide a private addressing alternative to using global addresses for intranet traffic. However, because the site-local address prefix can be used to address multiple sites within an organization, a site-local address prefix address can be duplicated. The ambiguity of site-local addresses in an organization adds complexity and difficulty for applications, routers, and network managers. For more information, see section 2 of RFC 3879.

To replace site-local addresses with a new type of address that is private to an organization, yet unique across all of the sites of the organization, RFC 4193 defines unique local IPv6 unicast addresses. Figure 4 shows the structure of unique local addresses.

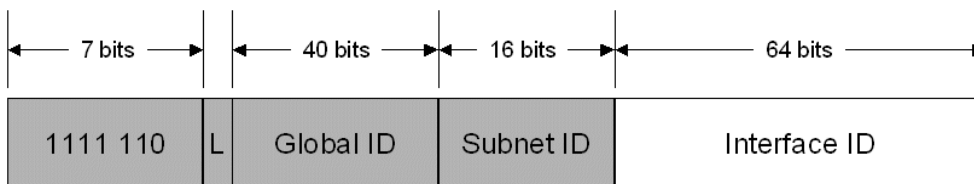


Figure 8 The unique local address

The first 7 bits have the fixed binary value of 1111110. All unique local addresses have the address prefix FC00::/7. The Local (L) flag is set 1 to indicate a local address. The L flag value set to 0 has not yet been defined. Therefore, unique local addresses with the L flag set to 1 have the address prefix of FD00::/8. The Global ID identifies a specific site within an organization and is set to a randomly derived 40-bit value. By deriving a random value for the Global ID, an organization can have statistically unique 48-bit prefixes assigned to the sites of their organizations. Additionally, two organizations that use unique local addresses that merge have a low probability of duplicating a 48-bit unique local address prefix, minimizing site renumbering. Unlike the Global Routing Prefix in global addresses, you should not assign Global IDs in unique local address prefixes so that they can be summarized.

The global address and unique local address share the same structure beyond the first 48 bits of the address. Figure 9 shows the structure of global and unique local addresses.

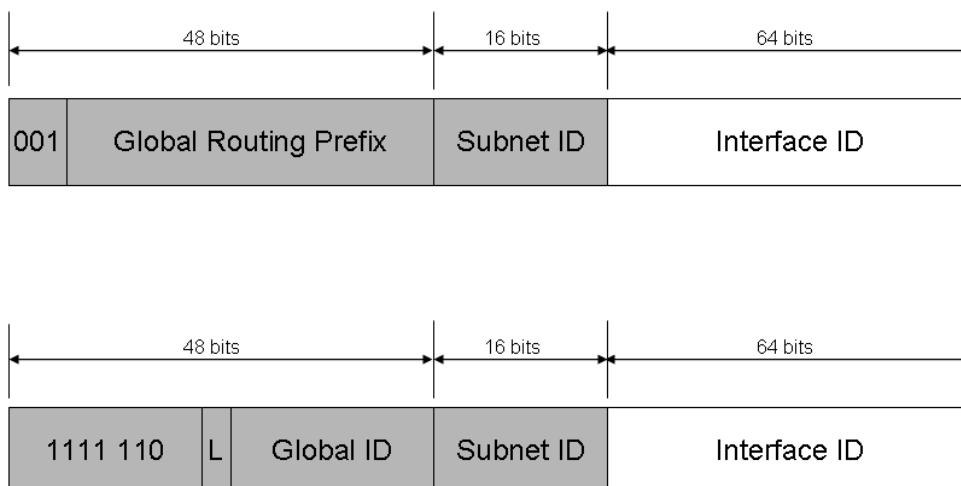


Figure 9 The structure of global and unique local addresses

In global addresses, the Subnet ID field identifies the subnet within an organization. For unique local addresses, the Subnet ID field can perform the same function. Therefore, you can create a subnet numbering scheme that can be used for both local and global unicast addresses.

Unique local addresses have a global scope but their reachability is defined by routing topology. Organizations will not advertise their unique local address prefixes outside of their organizations or create DNS AAAA entries with unique local addresses in the Internet DNS.

### Special IPv6 Addresses

The following are special IPv6 addresses:

- Unspecified address

The unspecified address (0:0:0:0:0:0:0:0 or ::) is only used to indicate the absence of an address. It is equivalent to the IPv4 unspecified address of 0.0.0.0. The unspecified address is typically used as a source address for packets attempting to verify the uniqueness of a tentative address. The unspecified address is never assigned to an interface or used as a destination address.

- Loopback address

The loopback address (0:0:0:0:0:0:0:1 or ::1) is used to identify a loopback interface, enabling a node to send packets to itself. It is equivalent to the IPv4 loopback address of 127.0.0.1. Packets addressed to the loopback address must never be sent on a link or forwarded by an IPv6 router.

### Compatibility Addresses

To aid in the migration from IPv4 to IPv6 and the coexistence of both types of hosts, the following addresses are defined:

- IPv4-compatible address

The IPv4-compatible address, 0:0:0:0:0:w.x.y.z or ::w.x.y.z (where w.x.y.z is the dotted decimal representation of an IPv4 address), is used by IPv6/IPv4 nodes that are communicating using IPv6. IPv6/IPv4 nodes are nodes with both IPv4 and IPv6 protocols. When the IPv4-compatible address

is used as an IPv6 destination, the IPv6 traffic is automatically encapsulated with an IPv4 header and sent to the destination using the IPv4 infrastructure.

- IPv4-mapped address

The IPv4-mapped address, `0:0:0:0:FFFF:w.x.y.z` or `::FFFF:w.x.y.z`, is used to represent an IPv4-only node to an IPv6 node. It is used only for internal representation. The IPv4-mapped address is never used as a source or destination address of an IPv6 packet.

- 6to4 address

The 6to4 address is used for communicating between two nodes running both IPv4 and IPv6 over an IPv4 routing infrastructure. The 6to4 address is formed by combining the prefix `2002::/16` with the 32 bits of a public IPv4 address, forming a 48-bit prefix. 6to4 is a tunneling technique described in RFC 3056.

For more information on these address and IPv6 transition technologies, see [IPv6 Transition Technologies](http://www.microsoft.com/technet/network/ipv6/ipv6coexist.mspx) at <http://www.microsoft.com/technet/network/ipv6/ipv6coexist.mspx>.

## Multicast IPv6 Addresses

In IPv6, multicast traffic operates in the same way that it does in IPv4. Arbitrarily located IPv6 nodes can listen for multicast traffic on an arbitrary IPv6 multicast address. IPv6 nodes can listen to multiple multicast addresses at the same time. Nodes can join or leave a multicast group at any time.

IPv6 multicast addresses have the first eight bits set to `1111 1111`. An IPv6 address is easy to classify as multicast because it always begins with “FF”. Multicast addresses cannot be used as source addresses or as intermediate destinations in a Routing extension header.

Beyond the first eight bits, multicast addresses include additional structure to identify their flags, scope, and multicast group. Figure 10 shows the IPv6 multicast address.

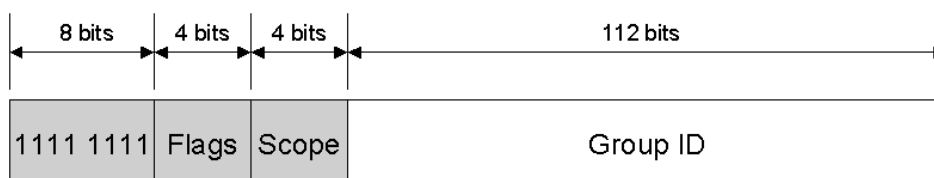


Figure 10 The IPv6 multicast address

The fields in the multicast address are:

- **Flags** – Indicates flags set on the multicast address. The size of this field is 4 bits. The first low-order bit is the Transient (T) flag. When set to 0, the T flag indicates that the multicast address is a permanently assigned (well-known) multicast address allocated by IANA. When set to 1, the T flag indicates that the multicast address is a transient (non-permanently-assigned) multicast address. The second low-order bit is for the Prefix (P) flag, which indicates whether the multicast address is based on a unicast address prefix. RFC 3306 describes the P flag. The third low-order bit is for the Rendezvous Point Address (R) flag, which indicates whether the multicast address contains an embedded rendezvous point address. RFC 3956 describes the R flag.
- **Scope** – Indicates the scope of the IPv6 internetwork for which the multicast traffic is intended. The size of this field is 4 bits. In addition to information provided by multicast routing protocols, routers use the

multicast scope to determine whether multicast traffic can be forwarded. The most prevalent values for the Scope field are 1 (interface-local scope), 2 (link-local scope), and 5 (site-local scope).

For example, traffic with the multicast address of FF02::2 has a link-local scope. An IPv6 router never forwards this traffic beyond the local link.

- **Group ID** – Identifies the multicast group and is unique within the scope. The size of this field is 112 bits. Permanently assigned group IDs are independent of the scope. Transient group IDs are only relevant to a specific scope. Multicast addresses from FF01:: through FF0F:: are reserved, well-known addresses.

To identify all nodes for the interface-local and link-local scopes, the following addresses are defined:

- FF01::1 (interface-local scope all-nodes multicast address)
- FF02::1 (link-local scope all-nodes multicast address)

To identify all routers for the interface-local, link-local, and site-local scopes, the following addresses are defined:

- FF01::2 (interface-local scope all-routers multicast address)
- FF02::2 (link-local scope all-routers multicast address)
- FF05::2 (site-local scope all-routers multicast address)

For the current list of permanently assigned IPv6 multicast addresses, see <http://www.iana.org/assignments/ipv6-multicast-addresses>.

### Solicited-Node Address

The solicited-node address facilitates the efficient querying of network nodes during address resolution. In IPv4, the ARP Request frame is sent to the MAC-level broadcast, disturbing all nodes on the network segment, including those that are not running IPv4. IPv6 uses the Neighbor Solicitation message to perform address resolution. However, instead of using the local-link scope all-nodes multicast address as the Neighbor Solicitation message destination, which would disturb all IPv6 nodes on the local link, the solicited-node multicast address is used. The solicited-node multicast address is comprised of the prefix FF02::1:FF00:0/104 and the last 24-bits of the IPv6 address that is being resolved, as shown in Figure 11.

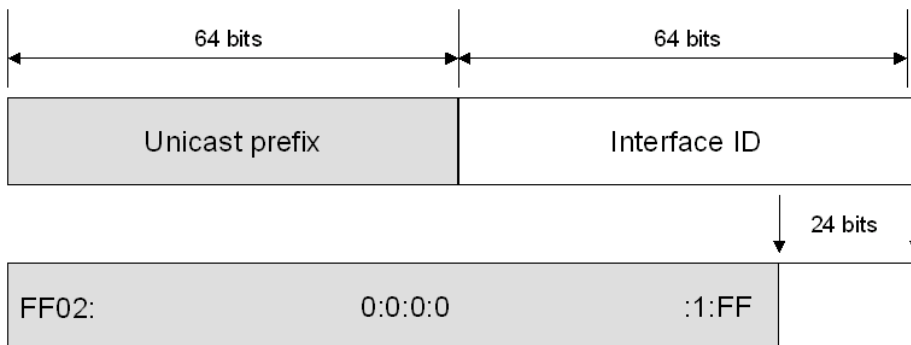


Figure 11 The solicited-node multicast address

For example, Node A is assigned the link-local address of FE80::2AA:FF:FE28:9C5A and is also listening on the corresponding solicited-node multicast address of FF02::1:FF28:9C5A (the underline highlights the correspondence of the last six hexadecimal digits). Node B on the local link must resolve Node A's link-local address FE80::2AA:FF:FE28:9C5A to its corresponding link-layer address. Node B sends a Neighbor Solicitation message to the solicited node multicast address of FF02::1:FF28:9C5A. Because Node A is listening on this multicast address, it processes the Neighbor Solicitation message and sends a unicast Neighbor Advertisement message in reply.

The result of using the solicited-node multicast address is that address resolutions, a common occurrence on a link, are not required to use a mechanism that disturbs all network nodes. By using the solicited-node address, very few nodes are disturbed during address resolution. In practice, due to the relationship between the Ethernet MAC address, the IPv6 interface ID, and the solicited-node address, the solicited-node address acts as a pseudo-unicast address for very efficient address resolution.

## Anycast IPv6 Addresses

An anycast address is assigned to multiple interfaces. Packets addressed to an anycast address are forwarded by the routing infrastructure to the nearest interface to which the anycast address is assigned. In order to facilitate delivery, the routing infrastructure must be aware of the interfaces assigned anycast addresses and their "distance" in terms of routing metrics. At present, anycast addresses are only used as destination addresses and are only assigned to routers. Anycast addresses are assigned out of the unicast address space and the scope of an anycast address is the scope of the type of unicast address from which the anycast address is assigned.

The Subnet-Router anycast address is predefined and required. It is created from the subnet prefix for a given interface. To construct the Subnet-Router anycast address, the bits in the subnet prefix are fixed at their appropriate values and the remaining bits are set to 0. All router interfaces attached to a subnet are assigned the Subnet-Router anycast address for that subnet. The Subnet-Router anycast address is used for communication with one of multiple routers attached to a remote subnet.

## IPv6 Addresses for a Host

An IPv4 host with a single network adapter typically has a single IPv4 address assigned to that adapter. An IPv6 host, however, usually has multiple IPv6 addresses—even with a single interface. An IPv6 host is assigned the following unicast addresses:

- A link-local address for each interface
- Unicast addresses for each interface (which could be a site-local address and one or multiple global unicast addresses)
- The loopback address (::1) for the loopback interface

Typical IPv6 hosts are logically multihomed because they have at least two addresses with which they can receive packets—a link-local address for local link traffic and a routable site-local or global address.

Additionally, each host is listening for traffic on the following multicast addresses:

- The interface-local scope all-nodes multicast address (FF01::1)
- The link-local scope all-nodes multicast address (FF02::1)
- The solicited-node address for each unicast address on each interface

- The multicast addresses of joined groups on each interface

## IPv6 Addresses for a Router

An IPv6 router is assigned the following unicast addresses:

- A link-local address for each interface
- Unicast addresses for each interface (which could be a site-local address and one or multiple global unicast addresses)
- A Subnet-Router anycast address
- Additional anycast addresses (optional)
- The loopback address (::1) for the loopback interface

Additionally, each router is listening for traffic on the following multicast addresses:

- The interface-local scope all-nodes multicast address (FF01::1)
- The interface-local scope all-routers multicast address (FF01::2)
- The link-local scope all-nodes multicast address (FF02::1)
- The link-local scope all-routers multicast address (FF02::2)
- The site-local scope all-routers multicast address (FF05::2)
- The solicited-node address for each unicast address on each interface
- The multicast addresses of joined groups on each interface

## IPv6 Interface Identifiers

The last 64 bits of an IPv6 address are the interface identifier that is unique to the 64-bit prefix of the IPv6 address. The following are the ways in which an IPv6 interface identifier is determined:

- A 64-bit interface identifier that is derived from the Extended Unique Identifier (EUI)-64 address. The 64-bit EUI-64 address is defined by the Institute of Electrical and Electronic Engineers (IEEE). EUI-64 addresses are either assigned to a network adapter or derived from IEEE 802 addresses. This is the default behavior for IPv6 in Windows XP and Windows Server 2003.
- As defined in RFC 4941, it might have a temporarily assigned, randomly generated interface identifier to provide a level of anonymity when acting as a client.
- As defined in RFC 5072, an interface identifier can be based on link-layer addresses or serial numbers, or randomly generated when configuring a Point-to-Point Protocol (PPP) interface and an EUI-64 address is not available.
- It is assigned during manual address configuration.
- It is a permanent interface identifier that is randomly generated to mitigate address scans of unicast IPv6 addresses on a subnet. This is the default behavior for IPv6 in Windows Vista and Windows Server 2008. You can disable this behavior with the **netsh interface ipv6 set global randomizeidentifiers=disabled** command.

## EUI-64 address-based interface identifiers

RFC 4291 states that all unicast addresses that use the prefixes 001 through 111 must also use a 64-bit interface identifier that is derived from the EUI-64 address. The 64-bit EUI-64 address is defined by the Institute of Electrical and Electronic Engineers (IEEE). EUI-64 addresses are either assigned to a network adapter card or derived from IEEE 802 addresses.

### IEEE 802 Addresses

Traditional interface identifiers for network adapters use a 48-bit address called an IEEE 802 address. It consists of a 24-bit company ID (also called the manufacturer ID), and a 24-bit extension ID (also called the board ID). The combination of the company ID, which is uniquely assigned to each manufacturer of network adapters, and the board ID, which is uniquely assigned to each network adapter at the time of assembly, produces a globally unique 48-bit address. This 48-bit address is also called the physical, hardware, or media access control (MAC) address.

Figure 12 shows the structure of the 48-bit IEEE 802 address.

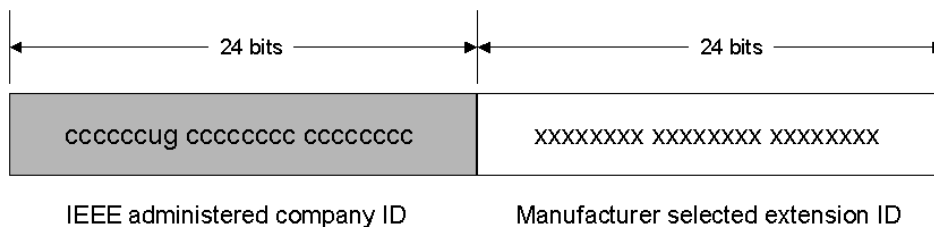


Figure 12 The 48-bit IEEE 802 address

Defined bits within the IEEE 802 address are:

**Universal/Local (U/L)** – The next-to-the low order bit in the first byte is used to indicate whether the address is universally or locally administered. If the U/L bit is set to 0, the IEEE (through the designation of a unique company ID) has administered the address. If the U/L bit is set to 1, the address is locally administered. The network administrator has overridden the manufactured address and specified a different address. The U/L bit is designated by the **u** in Figure 12.

**Individual/Group (I/G)** – The low order bit of the first byte is used to indicate whether the address is an individual address (unicast) or a group address (multicast). When set to 0, the address is a unicast address. When set to 1, the address is a multicast address. The I/G bit is designated by the **g** in Figure 12.

For a typical 802 network adapter address, both the U/L and I/G bits are set to 0, corresponding to a universally administered, unicast MAC address.

### IEEE EUI-64 Addresses

The IEEE EUI-64 address represents a new standard for network interface addressing and is used for Gigabit Ethernet adapters. The company ID is still 24-bits long, but the extension ID is 40 bits, creating a much larger address space for a network adapter manufacturer. The EUI-64 address uses the U/L and I/G bits in the same way as the IEEE 802 address.

Figure 13 shows the structure of the EUI-64 address.

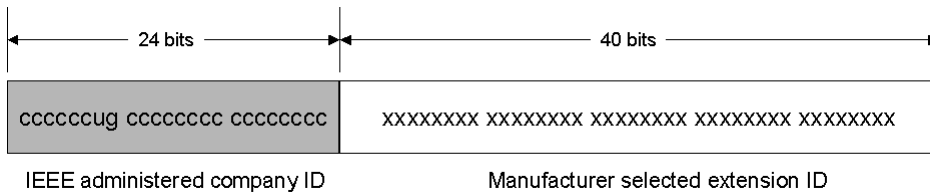


Figure 13 The EUI-64 address

**Mapping IEEE 802 Addresses to EUI-64 Addresses**

To create an EUI-64 address from an IEEE 802 address, the 16 bits of 11111111 11111110 (0xFFFE) are inserted into the IEEE 802 address between the company ID and the extension ID, as shown in Figure 14.

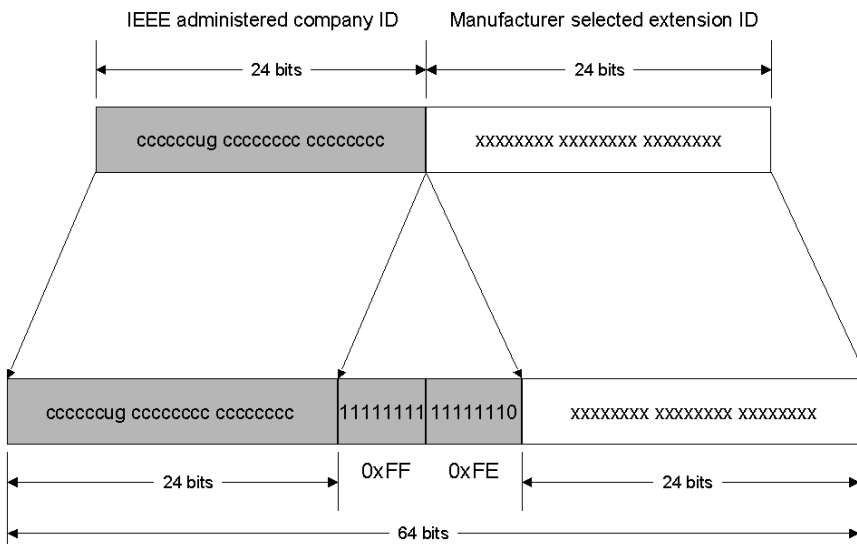


Figure 14 The conversion of an IEEE 802 address to an EUI-64 address

**Mapping EUI-64 Addresses to IPv6 Interface Identifiers**

To obtain the 64-bit interface identifier for IPv6 unicast addresses, the U/L bit in the EUI-64 address is complemented (if it is a 1, it is set to 0; and if it is a 0, it is set to 1). Figure 15 shows the conversion for a universally administered, unicast EUI-64 address.

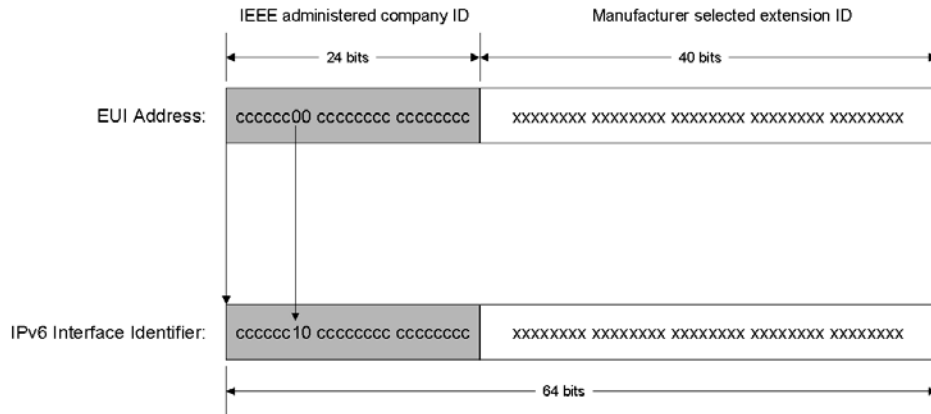


Figure 15 The conversion of a universally administered, unicast EUI-64 address to an IPv6 interface identifier

To obtain an IPv6 interface identifier from an IEEE 802 address, you must first map the IEEE 802 address to an EUI-64 address, and then complement the U/L bit. Figure 16 shows this conversion process for a universally administered, unicast IEEE 802 address.

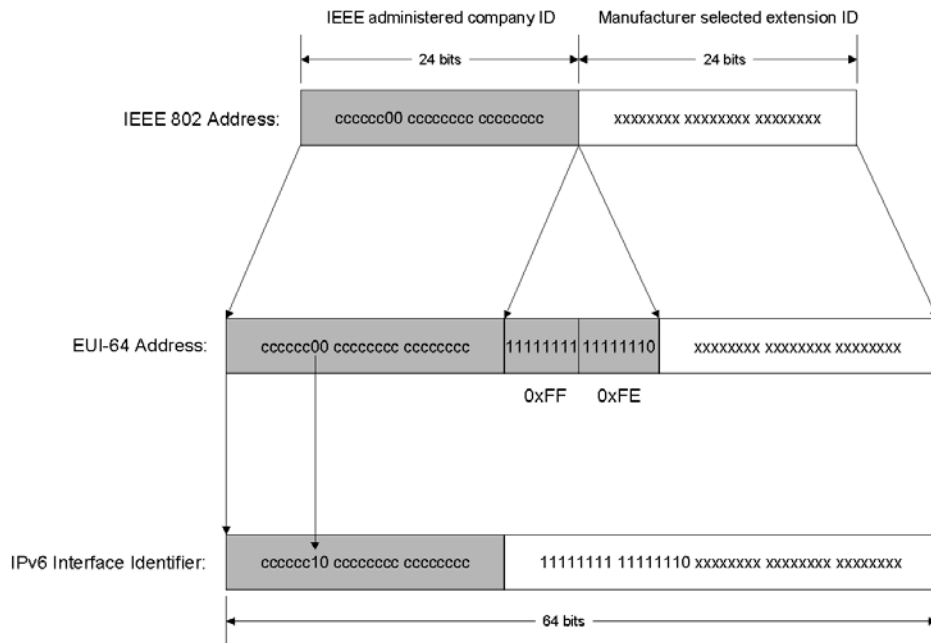


Figure 16 The conversion of a universally administered, unicast IEEE 802 address to an IPv6 interface identifier

**IEEE 802 Address Conversion Example**

Host A has the Ethernet MAC address of 00-AA-00-3F-2A-1C. First, it is converted to EUI-64 format by inserting FF-FE between the third and fourth bytes, yielding 00-AA-00-FF-FE-3F-2A-1C. Then, the U/L bit, which is the seventh bit in the first byte, is complemented. The first byte in binary form is 00000000. When the seventh bit is complemented, it becomes 00000010 (0x02). The final result is 02-AA-00-FF-FE-3F-2A-1C which, when converted to colon hexadecimal notation, becomes the interface identifier 2AA:FF:FE3F:2A1C. As a result, the link-local address that corresponds to the network adapter with the MAC address of 00-AA-00-2A-1C is FE80::2AA:FF:FE3F:2A1C.

**Note** When complementing the U/L bit, add 0x2 to the first byte if the address is universally administered, and subtract 0x2 from the

first byte if the address is locally administered.

### Temporary Address Interface Identifiers

In today's IPv4-based Internet, a typical Internet user connects to an Internet service provider (ISP) and obtains an IPv4 address using the Point-to-Point Protocol (PPP) and the Internet Protocol Control Protocol (IPCP). Each time the user connects, a different IPv4 address might be obtained. Because of this, it is difficult to track a dial-up user's traffic on the Internet on the basis of IP address.

For IPv6-based dial-up connections, the user is assigned a 64-bit prefix after the connection is made through router discovery and stateless address autoconfiguration. If the interface identifier is always based on the EUI-64 address (as derived from the static IEEE 802 address), it is possible to identify the traffic of a specific node regardless of the prefix, making it easy to track a specific user and their use of the Internet. To address this concern and provide a level of anonymity, an alternative IPv6 interface identifier that is randomly generated and changes over time is described in RFC 4941.

The initial interface identifier is generated by using random numbers. For IPv6 systems that cannot store any historical information for generating future interface identifier values, a new random interface identifier is generated each time the IPv6 protocol is initialized. For IPv6 systems that have storage capabilities, a history value is stored and, when the IPv6 protocol is initialized, a new interface identifier is created through the following process:

1. Retrieve the history value from storage and append the interface identifier based on the EUI-64 address of the adapter.
2. Compute the Message Digest-5 (MD5) one-way encryption hash over the quantity in step 1.
3. Save the last 64 bits of the MD5 hash computed in step 2 as the history value for the next interface identifier computation.
4. Take the first 64 bits of the MD5 hash computed in Step 2 and set the seventh bit to zero. The seventh bit corresponds to the U/L bit which, when set to 0, indicates a locally administered interface identifier. The result is the interface identifier.

The resulting IPv6 address, based on this random interface identifier, is known as a temporary address. Temporary addresses are generated for public address prefixes that use stateless address autoconfiguration. Temporary addresses are used for the lower of the following values of the valid and preferred lifetimes:

- The lifetimes included in the Prefix Information option in the received Router Advertisement message.
- Local default values of 1 week for valid lifetime and 1 day for preferred lifetime.

After the temporary address valid lifetime expires, a new interface identifier and temporary address is generated.

### Mapping IPv6 Multicast Addresses to Ethernet Addresses

When sending IPv6 multicast packets on an Ethernet link, the corresponding destination MAC address is 33-33-mm-mm-mm-mm where mm-mm-mm-mm is a direct mapping of the last 32 bits of the IPv6 multicast address, as shown in Figure 17.

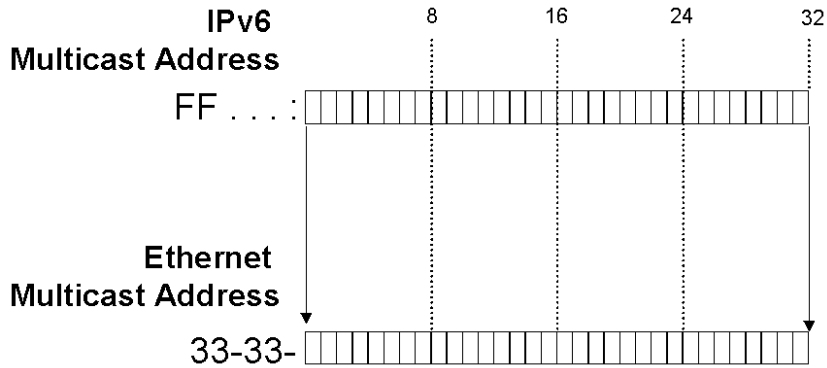


Figure 17 The mapping of an IPv6 multicast address to an Ethernet multicast MAC address

To efficiently receive IPv6 multicast packets on an Ethernet link, Ethernet network adapters can store additional interesting MAC addresses in a table on the network adapter. If an Ethernet frame with an interesting MAC address is received, it is passed to upper layers for additional processing. For every multicast address being listened to by the host, there is a corresponding entry in the table of interesting MAC address.

For example, a host with the Ethernet MAC address of 00-AA-00-3F-2A-1C (link-local address of FE80::2AA:FF:FE3F:2A1C) registers the following multicast MAC addresses with the Ethernet adapter:

- The address of 33-33-00-00-00-01, which corresponds to the link-local scope all-nodes multicast address of FF02::1.
- The address of 33-33-FF-3F-2A-1C, which corresponds to the solicited-node address of FF02::1:FF3F:2A1C. Remember that the solicited-node address is the prefix FF02::1:FF00:0/104 and the last 24-bits of the unicast IPv6 address.

Additional multicast addresses on which the host is listening are added and removed as needed from the table of interesting address on the Ethernet network adapter.

## IPv6 and DNS

Enhancements to the Domain Name System (DNS) for IPv6 are described in RFC 3596 and consist of the following new elements:

- Host address (AAAA) resource record
- IP6.ARPA domain for reverse queries

### The Host Address (AAAA) Resource Record

A new DNS resource record type, AAAA (called “quad A”), is used for resolving a fully qualified domain name to an IPv6 address. It is comparable to the host address (A) resource record used with IPv4. The resource record type is named AAAA (Type value of 28) because 128-bit IPv6 addresses are four times as large as 32-bit IPv4 addresses. The following is an example of a AAAA resource record:

```
host1.microsoft.com IN AAAA 2001:DB8:2F31:1A2D::2AA:FF:FE3F:2A1C
```

A host must specify either a AAAA query or a general query for a specific host name in order to receive IPv6 address resolution data in the DNS query answer sections.

## The IP6.ARPA Domain

The IP6.ARPA domain has been created for IPv6 reverse queries. Also called pointer queries, reverse queries determine a host name based on the IP address. To create the namespace for reverse queries, each hexadecimal digit in the fully expressed 32-digit IPv6 address becomes a separate level in inverse order in the reverse domain hierarchy.

For example, the reverse lookup domain name for the address FEC0::2AA:FF:FE3F:2A1C (fully expressed as FEC0:0000:0000:0000:02AA:00FF:FE3F:2A1C) is:

C.1.A.2.F.3.E.F.F.F.0.0.A.A.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.C.E.F.IP6.ARPA.

The DNS support described in RFC 3596 represents a simple way to both map host names to IPv6 addresses and provide reverse name resolution.

## Source and Destination Address Selection

For a typical IPv4-only host that has a single interface assigned one IPv4 address and resolves names using DNS, the choice of which IPv4 addresses to use as the source and destination when initiating communication is straightforward. The source IPv4 address is the address assigned to the interface of the host. The destination addresses to which connections are attempted are the IPv4 addresses returned in the DNS Name Query Response message.

For a typical IPv6 host that has multiple IPv6 addresses assigned to multiple interfaces and multiple IPv6 addresses are returned in the DNS Name Query Response message, the choice of the source and destination IPv6 address is more complex. The source and destination IPv6 addresses should be matched in scope and purpose. For example, an IPv6 host should not choose a link-local source address when communicating with a global destination address. Additionally, the possible destination address should be sorted by preference.

To provide a standardized method to choose source and destination IPv6 addresses with which to attempt connections, RFC 3484 defines the following required algorithms:

- A source address selection algorithm to choose the best source address to use with a destination address.
- A destination address selection algorithm to sort the list of possible destination addresses in order of preference.

For more information about the source and destination address selection algorithms defined in RFC 3484, see [Source and Destination Address Selection for IPv6](http://www.microsoft.com/technet/community/columns/cableguy/cg0206.mspx) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0206.mspx>.

## IPv4 Addresses and IPv6 Equivalents

Table 2 lists both IPv4 addresses and addressing concepts and their IPv6 equivalents.

**Table 2 IPv4 Addressing Concepts and Their IPv6 Equivalents**

IPv4 Address	IPv6 Address
Internet address classes	Not applicable in IPv6
Multicast addresses (224.0.0.0/4)	IPv6 multicast addresses (FF00::/8)
Broadcast addresses	Not applicable in IPv6

Unspecified address is 0.0.0.0	Unspecified address is ::
Loopback address is 127.0.0.1	Loopback address is ::1
Public IP addresses	Global unicast addresses
Private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16)	Site-local addresses (FEC0::/10)
Autoconfigured addresses (169.254.0.0/16)	Link-local addresses (FE80::/64)
Text representation: Dotted decimal notation	Text representation: Colon hexadecimal format with suppression of leading zeros and zero compression. IPv4-compatible addresses are expressed in dotted decimal notation.
Network bits representation: Subnet mask in dotted decimal notation or prefix length	Network bits representation: Prefix length notation only
DNS name resolution: IPv4 host address (A) resource record	DNS name resolution: IPv6 host address (AAAA) resource record
DNS reverse resolution: IN-ADDR.ARPA domain	DNS reverse resolution: IP6.ARPA domain

---

## IPv6 Header

The IPv6 header is a streamlined version of the IPv4 header. It eliminates fields that are unneeded or rarely used and adds fields that provide better support for real-time traffic.

### Structure of an IPv6 Packet

Figure 18 shows the structure of an IPv6 packet.

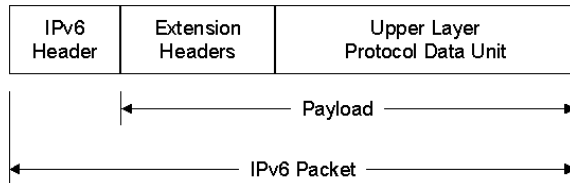


Figure 18 The structure of an IPv6 packet

### IPv6 Header

The IPv6 header is always present and is a fixed size of 40 bytes. The fields in the IPv6 header are described in detail later in this paper.

### Extension Headers

Zero or more extension headers can be present and are of varying lengths. A Next Header field in the IPv6 header indicates the next extension header. Within each extension header is a Next Header field that indicates the next extension header. The last extension header indicates the upper layer protocol (such as TCP, UDP, or ICMPv6) contained within the upper layer protocol data unit.

The IPv6 header and extension headers replace the existing IPv4 IP header with options. The new extension header format allows IPv6 to be augmented to support future needs and capabilities. Unlike options in the IPv4 header, IPv6 extension headers have no maximum size and can expand to accommodate all the extension data needed for IPv6 communication.

### Upper Layer Protocol Data Unit

The upper layer protocol data unit (PDU) usually consists of an upper layer protocol header and its payload (for example, an ICMPv6 message, a UDP message, or a TCP segment).

The IPv6 packet payload is the combination of the IPv6 extension headers and the upper layer PDU. Normally, it can be up to 65,535 bytes long. Payloads greater than 65,535 bytes in length can be sent using the Jumbo Payload option in the Hop-by-Hop Options extension header.

### IPv4 Header

A review of the IPv4 header is helpful in understanding the IPv6 header. Figure 19 shows the IPv4 header described in RFC 791.

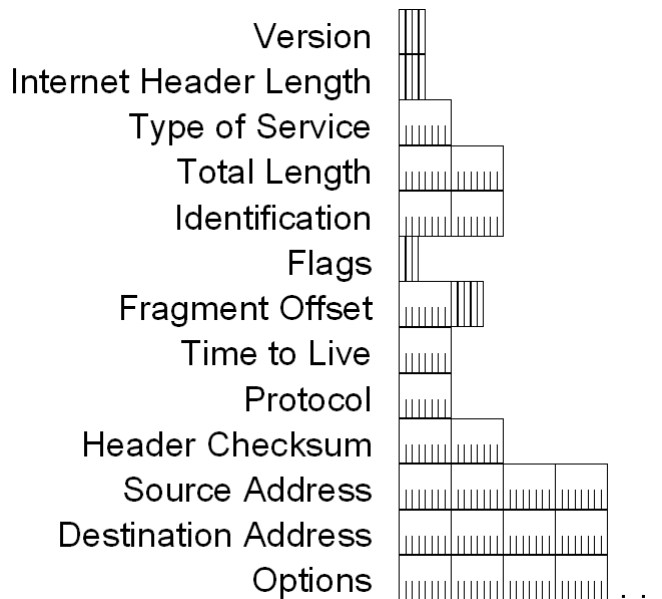


Figure 19 The IPv4 header

The fields in the IPv4 header are:

- **Version** – Indicates the version of IP and is set to 4. The size of this field is 4 bits.
- **Internet Header Length** – Indicates the number of 4-byte blocks in the IPv4 header. The size of this field is 4 bits. Because an IPv4 header is a minimum of 20 bytes in size, the smallest value of the Internet Header Length (IHL) field is 5. IPv4 options can extend the minimum IPv4 header size in increments of 4 bytes. If an IPv4 option does not use all 4 bytes of the IPv4 option field, the remaining bytes are padded with 0's, making the entire IPv4 header an integral number of 32-bits (4 bytes). With a maximum value of 0xF, the maximum size of the IPv4 header including options is 60 bytes (15×4).
- **Type of Service** – Renamed the Differentiated Services field in RFC 2472, this field indicates the desired service expected by this packet for delivery through routers across the IPv4 internetwork. The size of this field is 8 bits, which contains a 6-bit Differentiated Services Code Point (DSCP) field (RFC 2472) and two flags to support Explicit Congestion Notification (RFC 3168).
- **Total Length** – Indicates the total length of the IPv4 packet (IPv4 header + IPv4 payload) and does not include link layer framing. The size of this field is 16 bits, which can indicate an IPv4 packet that is up to 65,535 bytes long.
- **Identification** – Identifies this specific IPv4 packet. The size of this field is 16 bits. The Identification field is selected by the originating source of the IPv4 packet. If the IPv4 packet is fragmented, all of the fragments retain the Identification field value so that the destination node can group the fragments for reassembly.
- **Flags** – Identifies flags for the fragmentation process. The size of this field is 3 bits, however, only 2 bits are defined for current use. There are two flags—one to indicate whether the IPv4 packet might be fragmented and another to indicate whether more fragments follow the current fragment.
- **Fragment Offset** – Indicates the position of the fragment relative to the original IPv4 payload. The size of this field is 13 bits.

- **Time to Live** – Indicate the maximum number of links on which an IPv4 packet can travel before being discarded. The size of this field is 8 bits. The Time-to-Live field (TTL) was originally used as a time count with which an IPv4 router determined the length of time required (in seconds) to forward the IPv4 packet, decrementing the TTL accordingly. Modern routers almost always forward an IPv4 packet in less than a second and are required by RFC 791 to decrement the TTL by at least one. Therefore, the TTL becomes a maximum link count with the value set by the sending node. When the TTL equals 0, an ICMP Time Expired-TTL Expired in Transit message is sent to the source IPv4 address and the packet is discarded.
- **Protocol** – Identifies the upper layer protocol. The size of this field is 8 bits. For example, TCP uses a Protocol of 6, UDP uses a Protocol of 17, and ICMP uses a Protocol of 1. The Protocol field is used to demultiplex an IPv4 packet to the upper layer protocol.
- **Header Checksum** – Provides a checksum on the IPv4 header only. The size of this field is 16 bits. The IPv4 payload is not included in the checksum calculation as the IPv4 payload and usually contains its own checksum. Each IPv4 node that receives IPv4 packets verifies the IPv4 header checksum and silently discards the IPv4 packet if checksum verification fails. When a router forwards an IPv4 packet, it must decrement the TTL. Therefore, the Header Checksum is recomputed at each hop between source and destination.
- **Source Address** – Stores the IPv4 address of the originating host. The size of this field is 32 bits.
- **Destination Address** – Stores the IPv4 address of the destination host. The size of this field is 32 bits.
- **Options** – Stores one or more IPv4 options. The size of this field is a multiple of 32 bits. If the IPv4 option or options do not use all 32 bits, padding options must be added so that the IPv4 header is an integral number of 4-byte blocks that can be indicated by the Internet Header Length field.

## IPv6 Header

Figure 20 shows the IPv6 header as defined in RFC 2460.

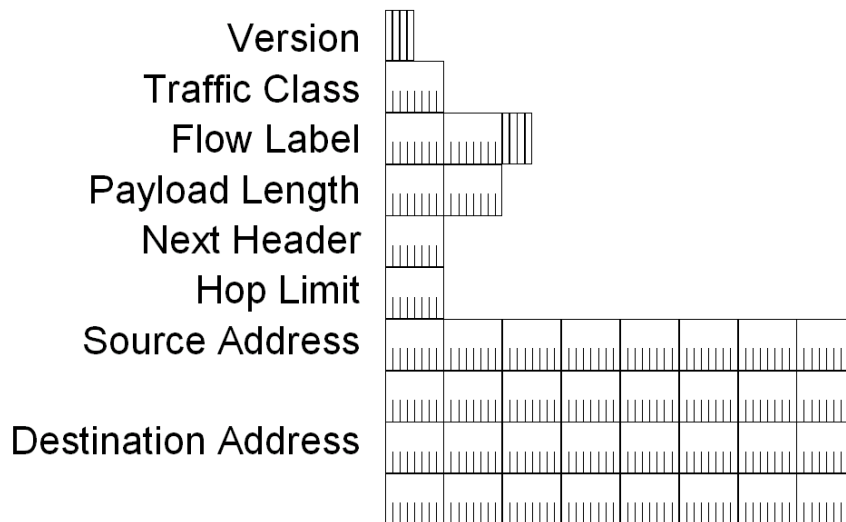


Figure 20 The IPv6 header

The fields in the IPv6 header are:

- **Version** – 4 bits are used to indicate the version of IP and is set to 6.
- **Traffic Class** – Like the IPv4 Type of Service field, specifies a DSCP values and flags for ECN. The size of this field is 8 bits.
- **Flow Label** – Indicates that this packet belongs to a specific sequence of packets between a source and destination, requiring special handling by intermediate IPv6 routers. The size of this field is 20 bits. The Flow Label is used for non-default quality of service connections, such as those needed by real-time data (voice and video). For default router handling, the Flow Label is set to 0. There can be multiple flows between a source and destination, as distinguished by separate non-zero Flow Labels. The use of the Flow Label field is defined in RFC 3697.
- **Payload Length** – Indicates the length of the IPv6 payload. The size of this field is 16 bits. The Payload Length field includes the extension headers and the upper layer PDU. With 16 bits, an IPv6 payload of up to 65,535 bytes can be indicated. For payload lengths greater than 65,535 bytes, the Payload Length field is set to 0 and the Jumbo Payload option is used in the Hop-by-Hop Options extension header.
- **Next Header** – Indicates either the first extension header (if present) or the protocol in the upper layer PDU (such as TCP, UDP, or ICMPv6). The size of this field is 8 bits. When indicating an upper layer protocol above the Internet layer, the same values used in the IPv4 Protocol field are used here.
- **Hop Limit** – Indicates the maximum number of links over which the IPv6 packet can travel before being discarded. The size of this field is 8 bits. The Hop Limit is similar to the IPv4 TTL field except that there is no historical relation to the amount of time (in seconds) that the packet is queued at the router. When the Hop Limit equals 0, an ICMPv6 Time Exceeded message is sent to the source address and the packet is discarded.
- **Source Address** – Stores the IPv6 address of the originating host. The size of this field is 128 bits.
- **Destination Address** – Stores the IPv6 address of the current destination host. The size of this field is 128 bits. In most cases the Destination Address is set to the final destination address. However, if a Routing extension header is present, the Destination Address might be set to the next router interface in the source route list.

### Values of the Next Header Field

Table 3 shows the typical values of the Next Header field for an IPv6 header or an IPv6 extension header.

Table 3 Values of the Next Header Field

Value (in decimal)	Header
0	Hop-by-Hop Options Header
6	TCP
17	UDP
41	Encapsulated IPv6 Header
43	Routing Header
44	Fragment Header
46	Resource ReSerVation Protocol

50	Encapsulating Security Payload
51	Authentication Header
58	ICMPv6
59	No next header
60	Destination Options Header

### Comparing the IPv4 and IPv6 Headers

Table 4 shows the differences between the IPv4 and IPv6 header fields.

Table 4 IPv4 Header Fields and Corresponding IPv6 Equivalents

IPv4 Header Field	IPv6 Header Field
Version	Same field but with different version numbers.
Internet Header Length	Removed in IPv6. IPv6 does not include a Header Length field because the IPv6 header is always a fixed size of 40 bytes. Each extension header is either a fixed size or indicates its own size.
Type of Service	Replaced by the IPv6 Traffic Class field.
Total Length	Replaced by the IPv6 Payload Length field, which only indicates the size of the payload.
Identification Fragmentation Flags Fragment Offset	Removed in IPv6. Fragmentation information is not included in the IPv6 header. It is contained in a Fragment extension header.
Time to Live	Replaced by the IPv6 Hop Limit field.
Protocol	Replaced by the IPv6 Next Header field.
Header Checksum	Removed in IPv6. In IPv6, bit-level error detection for the entire IPv6 packet is performed by the link layer.
Source Address	The field is the same except that IPv6 addresses are 128 bits in length.
Destination Address	The field is the same except that IPv6 addresses are 128 bits in length.
Options	Removed in IPv6. IPv4 options are replaced by IPv6 extension headers.

The one new field in the IPv6 header that is not included in the IPv4 header is the Flow Label field.

### IPv6 Extension Headers

The IPv4 header includes all options. Therefore, each intermediate router must check for their existence and process them when present. This can cause performance degradation in the forwarding of IPv4 packets. With IPv6, delivery and forwarding options are moved to extension headers. The only extension header that must be processed at each intermediate router is the Hop-by-Hop Options

extension header. This increases IPv6 header processing speed and improves forwarding process performance.

RFC 2460 defines the following IPv6 extension headers that must be supported by all IPv6 nodes:

- Hop-by-Hop Options header
- Destination Options header
- Routing header
- Fragment header
- Authentication header
- Encapsulating Security Payload header

In a typical IPv6 packet, no extension headers are present. If special handling is required by either the intermediate routers or the destination, one or more extension headers are added by the sending host.

Each extension header must fall on a 64-bit (8-byte) boundary. Extension headers of variable size contain a Header Extension Length field and must use padding as needed to ensure that their size is a multiple of 8 bytes.

Figure 21 shows the Next Header field in the IPv6 header and zero or more extension headers that form a chain of pointers. Each pointer indicates the type of header that comes after the immediate header until the upper layer protocol is ultimately identified.

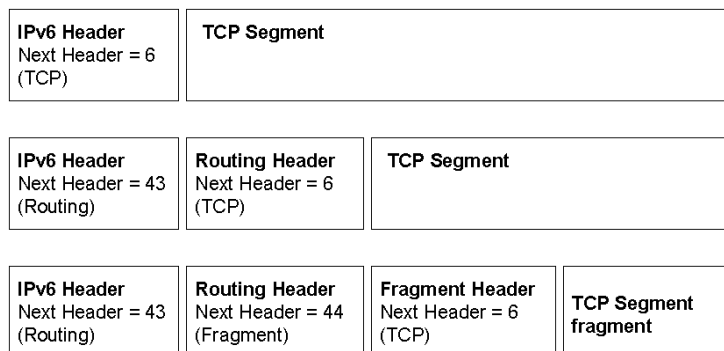


Figure 21 IPv6 extension headers

### Extension Headers Order

Extension headers are processed in the order in which they are present. Because the only extension header that is processed by every node on the path is the Hop-by-Hop Options header, it must be first. There are similar rules for other extension headers. In RFC 2460, it is recommended that extension headers be placed in the IPv6 header in the following order:

1. Hop-by-Hop Options header
2. Destination Options header (for intermediate destinations when the Routing header is present)
3. Routing header
4. Fragment header

5. Authentication header
6. Encapsulating Security Payload header
7. Destination Options header (for the final destination)

### Hop-by-Hop Options Header

The Hop-by-Hop Options header is used to specify delivery parameters at each hop on the path to the destination. It is identified by the value of 0 in the IPv6 header's Next Header field. Figure 22 shows the Hop-by-Hop Options header.

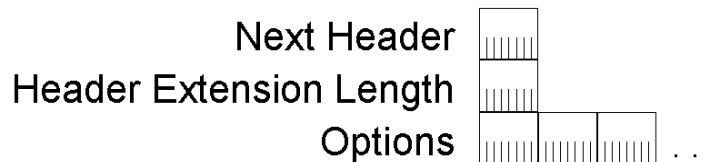


Figure 22 The Hop-by-Hop Options header

The Hop-by-Hop Options header consists of a Next Header field, a Header Extension Length field, and an Options field that contains one or more options. The value of the Header Extension Length field is the number of 8-byte blocks in the Hop-by-Hop Options extension header, not including the first 8 bytes. Therefore, for an 8-byte Hop-by-Hop Options header, the value of the Header Extension Length field is 0. Padding options are used to ensure 8-byte boundaries.

An option is a header within the Hop-by-Hop Options header that either describes a specific characteristic of the packet delivery or provides padding. Each option is encoded in the type-length-value (TLV) format that is commonly used in TCP/IP protocols. The option type both identifies the option and determines the way it is handled by the processing node. The option length identifies its length. The option value is the data associated with the option.

RFCs 2460, 2675, and 2711 define the following options:

- The Pad1 option (Option Type 0) is used to insert a single byte of padding.
- The PadN option (Option Type 1) is used to insert 2 or more bytes of padding.
- The Jumbo Payload option (Option Type 194) is used to indicate a payload size that is greater than 65,535 bytes. With the Jumbo Payload option, payload sizes of up to 4,294,967,295 bytes can be indicated using a 32-bit Jumbo Payload Length field. An IPv6 packet with a payload size greater than 65,535 bytes is named a *jumbogram*.
- The Router Alert option (Option Type 5) is used to indicate to the router that the contents of the packet require additional processing. The Router Alert option is used for Multicast Listener Discovery and the Resource ReSerVation Protocol (RSVP).

### Destination Options Header

The Destination Options header is used to specify packet delivery parameters for either intermediate destinations or the final destination. This header is identified by the value of 60 in the previous header's Next Header field. Figure 23 shows the Destination Options header.

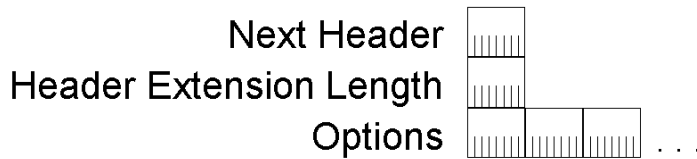


Figure 23 The Destination Options header

The fields within the Destination Options header are defined the same as the Hop-by-Hop Options header.

The Destination Options header is used in two ways:

1. If a Routing header is present, it specifies delivery or processing options at each intermediate destination.
2. It specifies delivery or processing options at the final destination.

### Routing Header

Similar to the loose source routing supported by IPv4, IPv6 source nodes can use the Routing extension header to specify a loose source route, a list of intermediate destinations for the packet to travel to on its path to the final destination. The Routing header is identified by the value of 43 in the previous header's Next Header field.

The Routing header consists of a Next Header field, a Header Extension Length field (defined the same way as the Hop-by-Hop Options extension header), a Routing Type field, a Segments Left field, and routing type-specific data.

For Routing Type 0, which is defined in RFC 2460, the routing type-specific data is a list of intermediate destination addresses. When the IPv6 packet reaches an intermediate destination, the Routing header is processed and the address of the next intermediate destination (based on the value of the Segments Left field) becomes the Destination Address in the IPv6 header.

Figure 24 shows the Routing header for Routing Type 0.

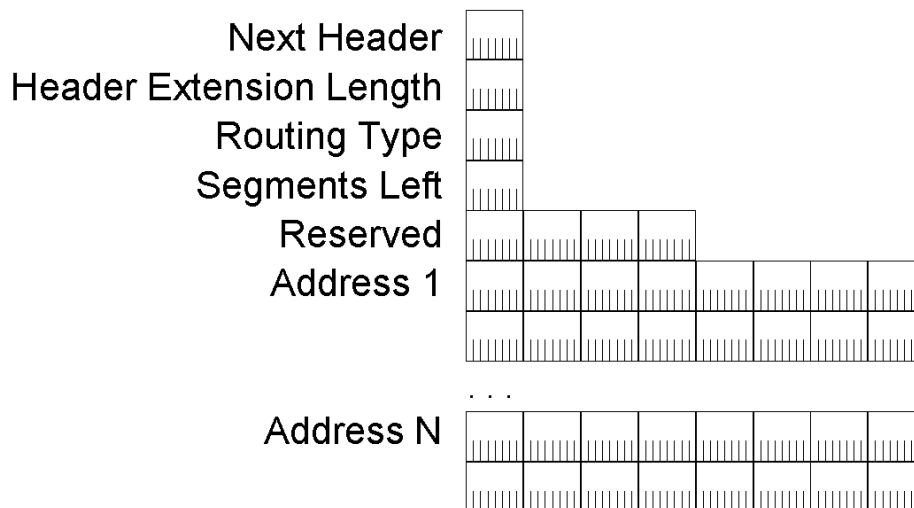


Figure 24 The Routing header for Routing Type 0

---

**Note** RFC 5095 formally deprecates the use of the Routing Type 0 extension header.

---

### Fragment Header

The Fragment header is used for IPv6 fragmentation and reassembly services. This header is identified by the value of 44 in the previous header's Next Header field. Figure 25 shows the Fragment header.

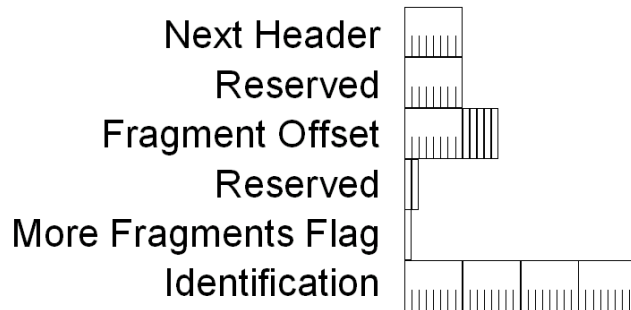


Figure 25 The Fragment header

The Fragment header includes a Next Header field, a 13-bit Fragment Offset field, a More Fragments flag, and a 32-bit Identification field. The Fragment Offset, More Fragments flag, and Identification fields are used in the same way as the corresponding fields in the IPv4 header. Because the use of the Fragment Offset field is defined for 8-byte fragment blocks, the Fragment header cannot be used for IPv6 jumbograms.

In IPv6, only source nodes can fragment payloads. If the payload submitted by the upper layer protocol is larger than the link or path MTU, then IPv6 fragments the payload at the source and uses the Fragment extension header to provide reassembly information.

When an IPv6 packet is fragmented, it is initially divided into unfragmentable and fragmentable parts:

- The unfragmentable part of the original IPv6 packet must be processed by each intermediate node between the fragmenting node and the destination. This part consists of the IPv6 header, the Hop-by-Hop Options header, the Destination Options header for intermediate destinations, and the Routing header.
- The fragmentable part of the original IPv6 packet must only be processed at the final destination node. This part consists of the Authentication header, the Encapsulating Security Payload header, the Destination Options header for the final destination, and the upper layer PDU.

Next, the IPv6 fragment packets are formed. Each fragment packet consists of the unfragmentable part, a fragment header, and a portion of the fragmentable part.

Figure 26 shows the fragmentation process for an IPv6 packet.

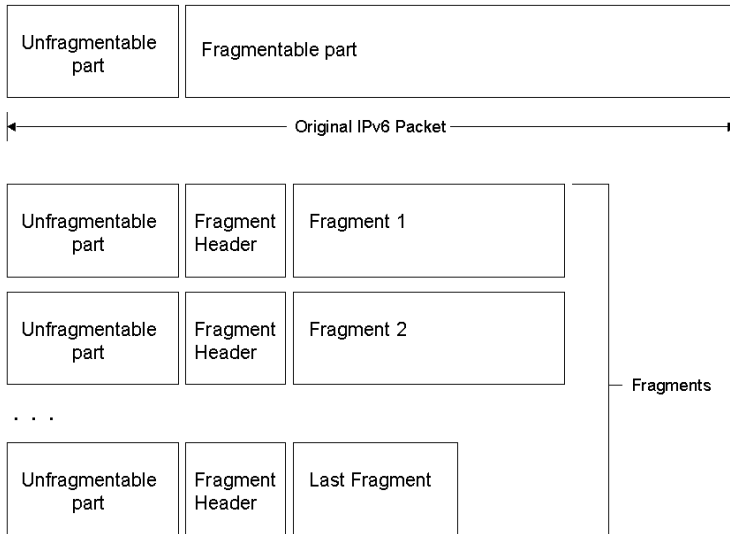


Figure 26 The IPv6 fragmentation process

### Authentication Header

The Authentication header provides data authentication (verification of the node that sent the packet), data integrity (verification that the data was not modified in transit), and anti-replay protection (assurance that captured packets cannot be retransmitted and accepted as valid data) for the IPv6 packet. The Authentication header, described in RFC 4302, is part of the Security Architecture for the Internet Protocol defined in RFC 4301.

The Authentication header is identified by the value of 51 in the previous header's Next Header field. Figure 27 shows the Authentication header.

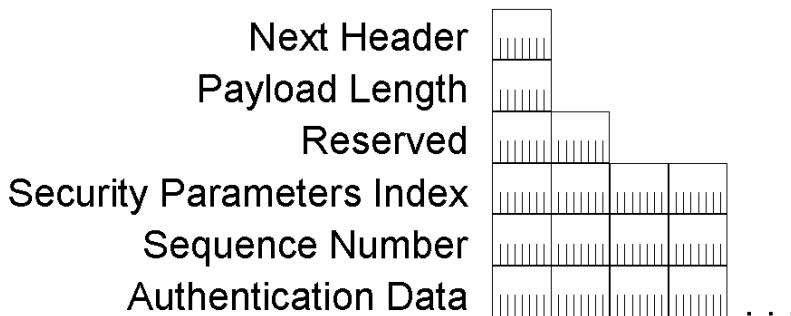


Figure 27 The Authentication header

The Authentication header contains a Next Header field, a Payload Length field, a Security Parameters Index (SPI) field that identifies a specific IP Security (IPsec) security association (SA), a Sequence Number field that provides anti-replay protection, and an Authentication Data field that contains an integrity check value (ICV). The ICV provides data authentication and integrity.

The Authentication extension header does not provide data confidentiality services by encrypting the data. To provide this, the Authentication header can be used in conjunction with the Encapsulating Security Payload (ESP) header.

Details about how the Authentication header provides data authentication and integrity through cryptographic techniques are beyond the scope of this paper. For more information, see RFC 4302.

### Encapsulating Security Payload Header and Trailer

The Encapsulating Security Payload (ESP) header and trailer provide data confidentiality, data authentication, and data integrity services to the encapsulated payload. In contrast, the Authentication header provides data authentication and integrity services for the entire IPv6 packet. The ESP header and trailer are identified by the value of 50 in the previous header's Next Header field. Figure 28 shows the ESP header and trailer.

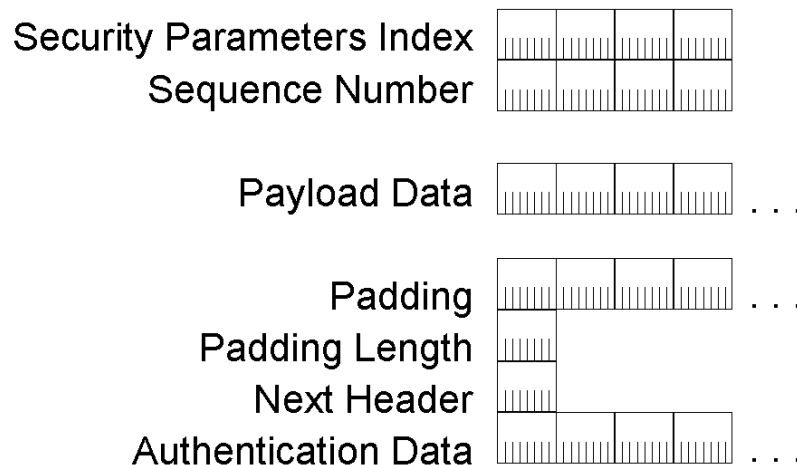


Figure 28 The ESP header and trailer

The ESP header contains a Security Parameters Index (SPI) field that identifies the IPsec SA and a Sequence Number field that provides anti-replay protection. The ESP trailer contains the Padding, Padding Length, Next Header, and Authentication Data fields. The Authentication Data field contains the integrity check value (ICV).

Details about how the ESP extension header and trailer provide data confidentiality, authentication, and integrity through cryptographic techniques are beyond the scope of this paper. For more information, see RFC 4303.

### IPv6 MTU

IPv6 requires that the link layer support a minimum IPv6 packet size of 1280 bytes. Link layers that do not support this must provide a link layer fragmentation and reassembly scheme that is transparent to IPv6. For link layers that can support a configurable MTU size, it is recommended that they be configured with an MTU size of at least 1500 bytes (the Ethernet II encapsulation IPv6 MTU). An example of a configurable MTU is the Maximum Receive Unit (MRU) of a Point-to-Point Protocol (PPP) link.

Like IPv4, IPv6 provides for a Path MTU Discovery process using the ICMPv6 Packet Too Big message described in "Path MTU Discovery." Path MTU Discovery allows the transmission of IPv6 packets for sizes greater than 1280 bytes.

IPv6 source hosts can fragment payloads of upper layer protocols that are larger than the path MTU by using the process and Fragment header previously described. However, the use of IPv6 fragmentation is highly discouraged. An IPv6 node must be able to reassemble a fragmented packet that is at least 1500 bytes in size.

### Upper Layer Checksums

The current implementation of TCP and UDP for IPv4 incorporates into their checksum calculation a pseudo-header that includes both the IPv4 Source Address and Destination Address fields. This checksum calculation must be modified for TCP and UDP traffic sent over IPv6 to include IPv6 addresses. Figure 29 shows the new pseudo-header that must be used by TCP and UDP checksums.

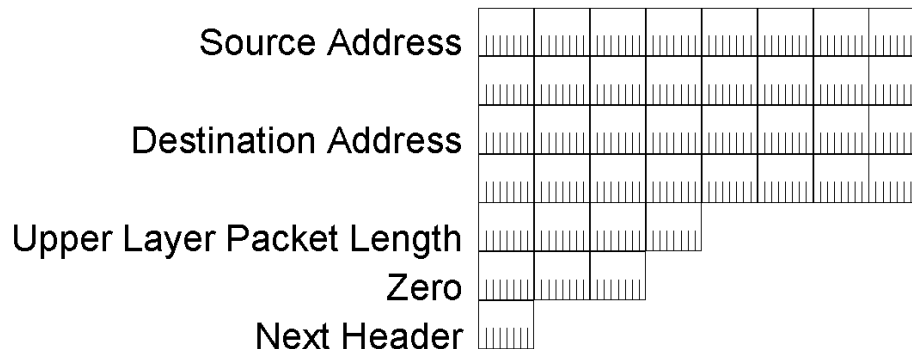


Figure 29 The IPv6 pseudo-header

The IPv6 pseudo-header includes the Source Address, the Destination Address, an Upper-Layer Packet Length field that indicates the length of the upper layer PDU, and a Next Header field that indicates the upper layer protocol for which the checksum is being calculated.

This pseudo-header is also used for the ICMPv6 checksum calculation.

---

## ICMPv6

Like IPv4, IPv6 does not provide facilities for reporting errors. Instead, IPv6 uses an updated version of the Internet Control Message Protocol (ICMP) named ICMP version 6 (ICMPv6). ICMPv6 has the common IPv4 ICMP functions of reporting delivery or forwarding errors and providing a simple echo service for troubleshooting.

The ICMPv6 protocol also provides a framework for the following:

- Multicast Listener Discovery (MLD)

MLD is a series of three ICMPv6 messages that replace version 2 of the Internet Group Management Protocol (IGMP) for IPv4 to manage subnet multicast membership. MLD is described in more detail in “Multicast Listener Discovery.”

- Neighbor Discovery (ND)

Neighbor Discovery is a series of five ICMPv6 messages that manage node-to-node communication on a link. Neighbor Discovery replaces Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and the ICMPv4 Redirect message. Neighbor Discovery is described in more detail in “Neighbor Discovery.”

ICMPv6 is required for an IPv6 implementation and is defined in RFC 4443.

### Types of ICMPv6 Messages

There are two types of ICMPv6 messages:

1. Error messages

Error messages are used to report errors in the forwarding or delivery of IPv6 packets by either the destination node or an intermediate router. The value of the 8-bit Type field in ICMPv6 error messages is in the range of 0 through 127 (the high order bit is set to 0). ICMPv6 error messages include Destination Unreachable, Packet Too Big, Time Exceeded, and Parameter Problem.

2. Informational messages

Informational messages are used to provide diagnostic functions and additional host functionality such as MLD and Neighbor Discovery. The value of the Type field in ICMPv6 informational messages is in the range of 128 through 255 (the high order bit is set to 1). ICMPv6 informational messages are described in RFC 4443 and include Echo Request and Echo Reply.

### ICMPv6 Header

An ICMPv6 header is indicated by setting the previous header's Next Header field to 58. Figure 30 shows the structure of all ICMPv6 messages.

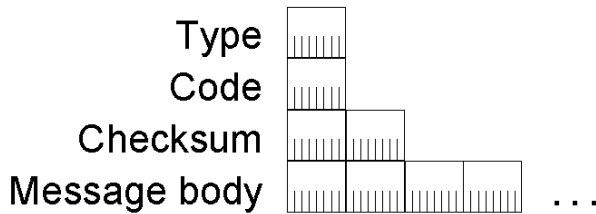


Figure 30 The ICMPv6 message structure

The fields in the ICMPv6 header are:

- **Type** – Indicates the type of ICMPv6 message. The size of this field is 8 bits. In ICMPv6 error messages, the high-order bit is set to 0. In ICMPv6 informational messages, the high-order bit is set to 1.
- **Code** – Differentiates among multiple messages within a given message type. The size of this field is 8 bits. If there is only one message for a given type, the Code field is set to 0.
- **Checksum** – Stores a checksum of the ICMPv6 message. The size of this field is 16 bits. The IPv6 pseudo-header is added to the ICMPv6 message when calculating the checksum.
- **Message body** – Contains ICMPv6 message-specific data.

### ICMPv6 Error Messages

ICMPv6 error messages are used to report forwarding or delivery errors by either a router or the destination host. To conserve network bandwidth, ICMPv6 error messages are not sent for every error encountered. Instead, ICMPv6 error messages are rate limited. Rate limiting reduces the amount of bandwidth consumed by reporting errors.

#### Destination Unreachable

An ICMPv6 Destination Unreachable message is sent by either a router or a destination host when the packet cannot be forwarded to its destination. Figure 31 shows the ICMPv6 Destination Unreachable message.

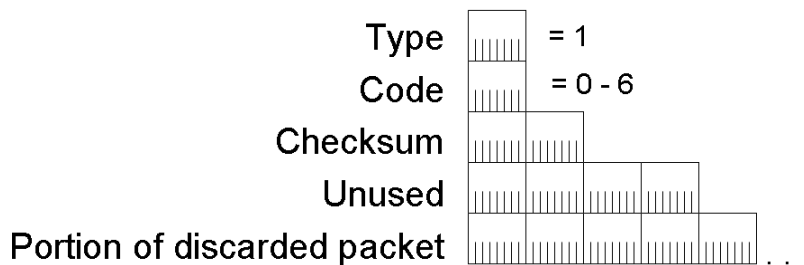


Figure 31 The ICMPv6 Destination Unreachable message

In the Destination Unreachable message, the Type field is set to 1 and the Code field is set to a value in the range of 0 through 4. After the Checksum field is the 32-bit Unused field and the portion of the discarded packet that makes the entire IPv6 packet containing the ICMPv6 message no larger than 1280 bytes (the minimum IPv6 MTU). The number of bytes of the discarded packet included in the message varies if there are IPv6 extension headers present. For an ICMPv6 message without

extension headers, 1232 bytes of the discarded packet are included (1280 less a 40-byte IPv6 header and an 8-byte ICMPv6 Destination Unreachable header).

Table 5 shows the value of the Code field for the various Destination Unreachable messages.

Table 5 ICMPv6 Destination Unreachable Messages

Code value	Description
0	No route matching the destination was found in the routing table.
1	The communication with the destination is prohibited by administrative policy. This is typically sent when the packet is discarded by a firewall.
2	The address is beyond the scope of the source address.
3	The destination address is unreachable. This is typically sent because of an inability to resolve the destination's link layer address.
4	The destination port was unreachable. This is typically sent when an IPv6 packet containing a UDP message arrived at the destination but there were no applications listening on the destination UDP port.
5	The packet with this source address is not allowed due to inbound (ingress) or outbound (egress) packet filtering policies.
6	The packet matched a reject route and was discarded. A reject route is an address prefix configured on a router for traffic that the router must immediately discard.

### Packet Too Big

An ICMPv6 Packet Too Big message is sent when the packet cannot be forwarded because the link MTU on the forwarding link is smaller than the size of the IPv6 packet. Figure 32 shows the ICMPv6 Packet Too Big message.

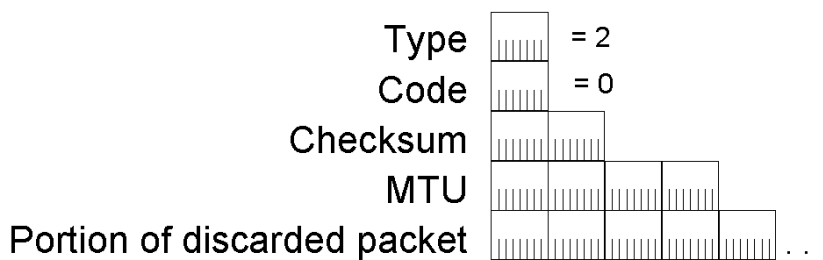


Figure 32 The ICMPv6 Packet Too Big message

In the Packet Too Big message, the Type field is set to 2 and the Code field is set to 0. After the Checksum field is the 32-bit MTU field that stores the link MTU for the link on which the packet was being forwarded. Next is the portion of the discarded packet that makes the entire IPv6 packet containing the ICMPv6 message no larger than the maximum length of 1280 bytes. The Packet Too Big message is used for the IPv6 Path MTU Discovery process described in “Path MTU Discovery.”

**Time Exceeded**

An ICMPv6 Time Exceeded message is typically sent by a router when the Hop Limit field in the IPv6 header is zero, either upon receipt or after decrementing its value during the forwarding process. Figure 33 shows the ICMPv6 Time Exceeded message.

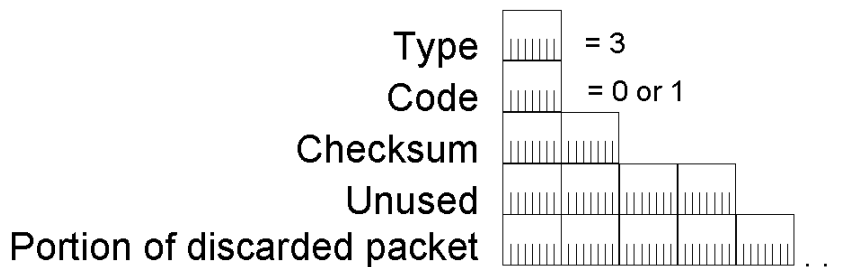


Figure 33 The ICMPv6 Time Exceeded message

In the Time Exceeded message, the Type field is set to 3 and the Code field is set to either 0 (when the Hop Limit field in the IPv6 header becomes 0) or 1 (when the fragmentation reassembly time of the destination host is exceeded). After the Checksum field is the 32-bit Unused field and the portion of the discarded packet that makes the entire IPv6 packet containing the ICMPv6 message no larger than 1280 bytes. The receipt of Time Exceeded messages for Code=0 indicates that either the Hop Limit of outgoing packets is not large enough to reach the destination or that a routing loop exists.

**Parameter Problem**

An ICMPv6 Parameter Problem message is either sent by a router or by the destination. This occurs when an error is encountered in either the IPv6 header or an extension header, preventing IPv6 from performing further processing. Figure 34 shows the ICMPv6 Parameter Problem message.

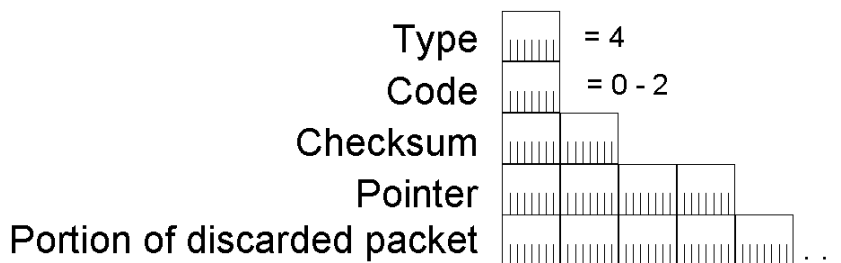


Figure 34 The ICMPv6 Parameter Problem message

In the Parameter Problem message, the Type field is set to 4 and the Code field is a value in the range of 0 through 2. After the Checksum field is the 32-bit Pointer field that indicates the byte offset in the offending IPv6 packet where the error was encountered. After the Pointer field is the portion of the discarded packet that makes the entire ICMPv6 message no larger than 1280 bytes. The Pointer field

value is set to the correct offset even when the location of the error is not included in the portion of the discarded packet.

Table 6 shows the Code field values for Parameter Problem messages.

Table 6 ICMPv6 Parameter Problem Messages

Code value	Description
0	An error in a field within the IPv6 header or an extension header was encountered.
1	An unrecognized Next Header field value was encountered. This is equivalent to the IPv4 Destination Unreachable-Protocol Unreachable message.
2	An unrecognized IPv6 option was encountered.

### ICMPv6 Informational Messages

ICMPv6 informational messages, defined in RFC 4443, provide diagnostic capabilities to aid in troubleshooting.

#### Echo Request

An ICMPv6 Echo Request message is sent to a destination to solicit an immediate Echo Reply message. The Echo Request/Echo Reply message facility provides a simple diagnostics function to aid in the troubleshooting of a variety of reachability and routing problems. Figure 35 shows the ICMPv6 Echo Request message.

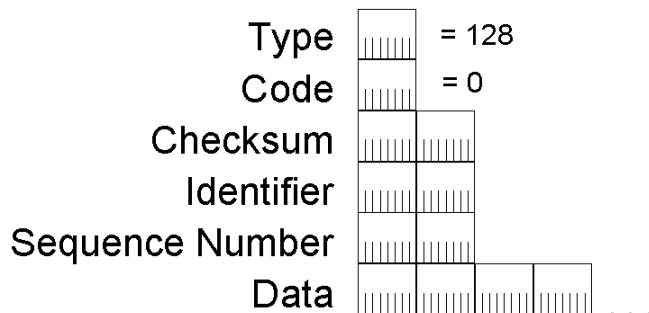


Figure 35 The ICMPv6 Echo Request message

In the Echo Request message, the Type field is set to 128 and the Code field is set to 0. After the Checksum field are the 16-bit Identifier and Sequence Number fields. The Identifier and Sequence Number fields are set by the sending host and used to match an incoming Echo Reply message with its corresponding Echo Request. The Data field is zero or more bytes of optional data that is also set by the sending host.

#### Echo Reply

An ICMPv6 Echo Reply message is sent in response to the receipt of an ICMPv6 Echo Request message. Figure 36 shows the ICMPv6 Echo Reply message.

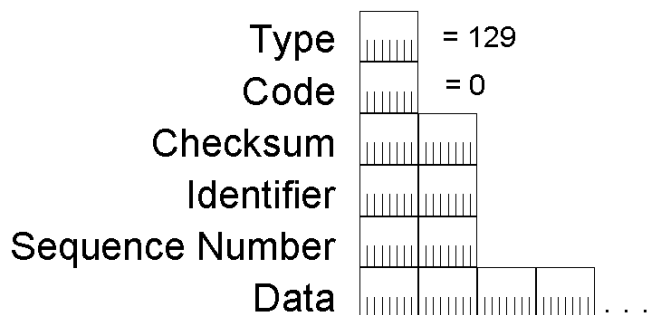


Figure 36 The ICMPv6 Echo Reply message

In the Echo Reply message, the Type field is set to 129 and the Code field is set to 0. After the Checksum field are the 16-bit Identifier and Sequence Number fields. The Identifier, Sequence Number, and Data fields are set with the same values as those in the Echo Request message that initially prompted the Echo Reply.

### Comparing ICMPv4 and ICMPv6 Error Messages

Table 7 lists commonly used ICMPv4 error messages and their corresponding ICMPv6 equivalents.

Table 7 ICMPv4 Error Messages and Their Corresponding ICMPv6 Equivalents

ICMPv4 Message	ICMPv6 Equivalent
Destination Unreachable-Network unreachable (Type 3, Code 1)	Destination Unreachable-No route to destination (Type 1, Code 0)
Destination Unreachable-Host unreachable (Type 3, Code 1)	Destination Unreachable-Address unreachable (Type 1, Code 3)
Destination Unreachable-Protocol unreachable (Type 3, Code 2)	Parameter Problem-Unrecognized Next Header field (Type 4, Code 1)
Destination Unreachable-Port unreachable (Type 3, Code 3)	Destination Unreachable-Port unreachable (Type 1, Code 4)
Destination Unreachable-Fragmentation needed and DF set (Type 3, Code 4)	Packet Too Big (Type 2, Code 0)
Destination Unreachable-Communication with destination host administratively prohibited (Type 3, Code 10)	Destination Unreachable-Communication with destination administratively prohibited (Type 1, Code 1)
Time Exceeded-TTL expired in transit (Type 11, Code 0)	Time Exceeded-Hop Limit exceeded (Type 3, Code 0)
Time Exceeded-Fragmentation timer expired (Type 11, Code 1)	Time Exceeded-Fragmentation timer exceeded (Type 3, Code 1)
Parameter Problem (Type 12, Code 0)	Parameter Problem (Type 4, Code 0 or Code 2)
Source Quench (Type 4, Code 0)	This message is not present in IPv6.
Redirect (Type 5, Code 0)	Neighbor Discovery Redirect message (Type 137, Code 0). For more information, see "Neighbor Discovery."

## Path MTU Discovery

The path MTU is the smallest link MTU of any link in the path between a source and a destination. IPv6 packets with a maximum size of the path MTU do not require fragmentation by the host and will be successfully forwarded by all routers on the path. To discover the path MTU, the sending node uses the receipt of ICMPV6 Packet Too Big messages.

The path MTU is discovered through the following process:

1. The sending node assumes that the path MTU is the link MTU of the interface on which the traffic is being forwarded.
2. The sending node sends IPv6 packets at the path MTU size.
3. If a router on the path is unable to forward the packet over a link with a link MTU that is smaller than the size of the packet, it discards the IPv6 packet and sends an ICMPV6 Packet Too Big message back to the sending node. The ICMPV6 Packet Too Big message contains the link MTU of the link on which the forwarding failed.
4. The sending node sets the path MTU for packets being sent to the destination to the value of the MTU field in the ICMPv6 Packet Too Big message.

The sending node starts again at step 2 and repeats steps 2 through 4 for as many times as are necessary to discover the path MTU. The path MTU is determined when either no additional ICMPv6 Packet Too Big messages are received or an acknowledgment is received from the destination.

In RFC 1981, it is recommended that IPv6 nodes support path MTU discovery. Those that do not must use the minimum link MTU of 1280 bytes as the path MTU.

### Changes in Path MTU

Due to changes in routing topology, the path between source and destination might change over time. When a new path requires a lower path MTU, the earlier process begins at step 3 and repeats steps 2 through 4 until the new path MTU is discovered.

Decreases in path MTU are immediately discovered through the receipt of ICMPV6 Packet Too Big messages. Increases in path MTU must be detected by the sending node. As described in RFC 1981, the sending node can attempt to send a larger IPv6 packet after a minimum of 5 minutes (10 minutes are recommended) upon receiving an ICMPv6 Packet Too Big message.

## Multicast Listener Discovery

Multicast Listener Discovery (MLD) is the IPv6 equivalent of Internet Group Management Protocol version 2 (IGMPv2) for IPv4. MLD is a set of messages exchanged by routers and nodes, enabling routers to discover the set of multicast addresses for which there are listening nodes for each attached interface. Like IGMPv2, MLD only discovers the list of multicast addresses for which there is at least one listener, not the list of individual multicast listeners for each multicast address. Multicast Listener Discovery (MLD) is documented in RFC 2710.

### MLD Messages

Unlike IGMPv2, MLD uses ICMPv6 messages instead of defining its own message structure. All MLD messages are ICMPv6 messages types 130, 131, and 132. The three types of MLD messages are:

#### 1. Multicast Listener Query

Multicast Listener Query is used by a router to query a link for multicast listeners. There are two types of Multicast Listener Query messages: The General Query and the Multicast-Address-Specific Query. The General Query is used to query for multicast listeners of all multicast addresses. The Multicast-Address-Specific Query is used to query for multicast listeners of a specific multicast address. The two message types are distinguished by the multicast destination address in the IPv6 header and a multicast address within the Multicast Listener Query message.

#### 2. Multicast Listener Report

Multicast Listener Report is used by a multicast listener to either report interest in receiving multicast traffic for a specific multicast address or to respond to a Multicast Listener Query.

#### 3. Multicast Listener Done

Multicast Listener Done is used by a multicast listener to report that it is no longer interested in receiving multicast traffic for a specific multicast address.

An MLD message packet consists of an IPv6 header, a Hop-by-Hop Options extension header, and the MLD message. The Hop-by-Hop Options extension header contains the IPv6 Router Alert Option documented in RFC 2711. It is used to ensure that routers process MLD messages sent to multicast addresses on which the router is not listening. Figure 37 shows the format of an MLD message packet.

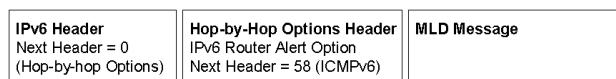


Figure 37 The Format of an MLD message packet

### Multicast Listener Query

An MLD Multicast Listener Query message is equivalent to the IGMPv2 Host Membership Query message. It is used by a router to query an attached link for listening hosts.

In the IPv6 header, the source address is the link-local address of the interface on which the query is being sent. The Hop Limit field is set to 1. For the General Query, the destination address is the link-local scope all-nodes multicast address (FF02::1). For the Multicast-Address-Specific Query, the destination address is the specific multicast address being queried.

Figure 38 shows the MLD Multicast Listener Query message.

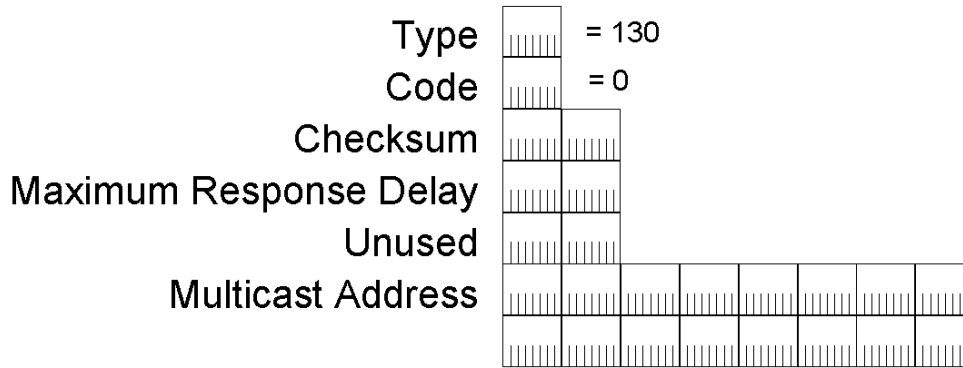


Figure 38 The MLD Multicast Listener Query message

In the MLD Multicast Listener Query message, the Type field is set to 130 and the Code field is set to 0. After the Checksum field are the 16-bit Maximum Response Delay and Reserved fields. The Maximum Response Delay is the maximum amount of time in milliseconds within which a multicast group member must report its membership using an MLD Multicast Listener Report message. In the General Query, the Multicast Address field is set to the unspecified address (::). In the Multicast-Address-Specific Query, the Multicast Address field is set to the specific multicast address that is being queried.

### Multicast Listener Report

An MLD Multicast Listener Report message is equivalent to the IGMPv2 Host Membership Report message. It is used by a listening node to either report its interest in receiving multicast traffic at a specific multicast address or respond to an MLD General or Multicast-Address-Specific Query message.

In the IPv6 header, the source address is the link-local address of the interface on which the report is being sent. The Hop Limit field is set to 1 and the destination address is the specific multicast address being reported.

Figure 39 shows the MLD Multicast Listener Report message.

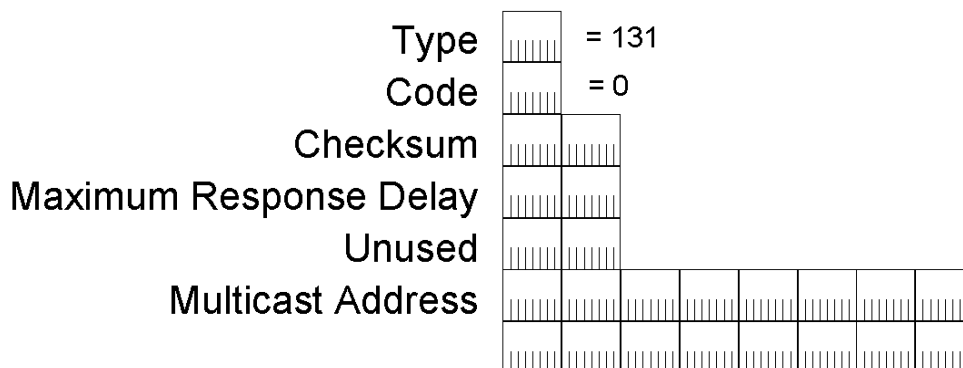


Figure 39 The MLD Multicast Listener Report message

In the MLD Multicast Listener Report message, the Type field is set to 131 and the Code field is set to 0. The Maximum Response Delay field is not used in a Multicast Listener Report message and is set to 0. The Multicast Address field is set to the specific multicast address that is being reported.

### Multicast Listener Done

An MLD Multicast Listener Done message is equivalent to the IGMPv2 Leave Group message. It is used by a multicast group member to inform local routers that it might be the last group member on the subnet.

In the IPv6 header, the source address is the link-local address of the interface on which the report is being sent. The Hop Limit field is set to 1 and the destination address is the link-local scope all-routers multicast address (FF02::2).

Figure 40 shows the MLD Multicast Listener Done message.

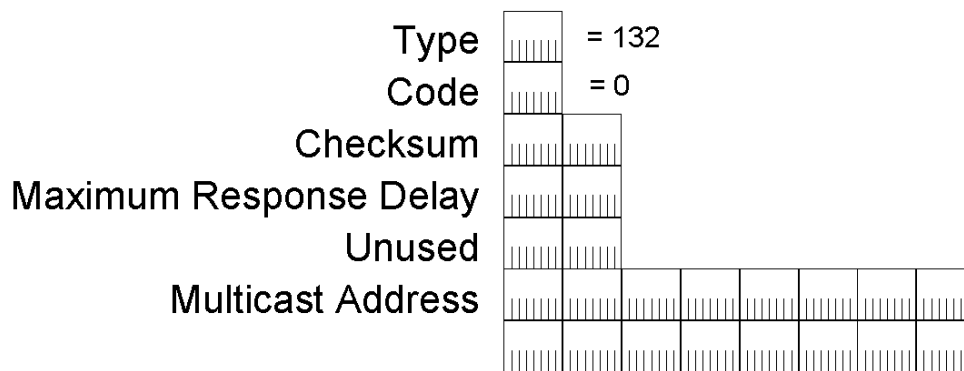


Figure 40 The MLD Multicast Listener Done message

In the MLD Multicast Listener Done message, the Type field is set to 132 and the Code field is set to 0. The Maximum Response Delay field is not used in a Multicast Listener Done message and is set to 0. The Multicast Address field is set to the specific multicast address for which the sending node is informing local routers that it is no longer as listener.

### MLDv2

Windows Vista and Windows Server 2008 also support Multicast Listener Discovery version 2 (MLDv2), specified in RFC 3810, which allows IPv6 hosts to register interest in source-specific multicast traffic with their neighboring routers. A host running Windows Vista or Windows Server 2008 can register interest in receiving IPv6 multicast traffic from only specific source addresses (an include list) or from any source except specific source addresses (an exclude list).

## Neighbor Discovery

IPv6 Neighbor Discovery (ND) is a set of messages and processes that determine relationships between neighboring nodes. ND replaces ARP, ICMP Router Discovery, and ICMP Redirect used in IPv4 and provides additional functionality.

ND is used by hosts to:

- Discover neighboring routers.
- Discover addresses, address prefixes, and other configuration parameters.

ND is used by routers to:

- Advertise their presence, host configuration parameters, and on-link prefixes.
- Inform hosts of a better next-hop address to forward packets for a specific destination.

ND is used by nodes to:

- Resolve the link-layer address of a neighboring node to which an IPv6 packet is being forwarded and determine when the link-layer address of a neighboring node has changed.
- Determine whether a neighbor is still reachable.

Table 8 lists and describes the ND processes documented in RFC 4861.

Table 8 IPv6 Neighbor Discovery Processes

Process	Description
Router discovery	The process by which a host discovers the local routers on an attached link. Equivalent to ICMPv4 Router Discovery. For more information, see "Router Discovery."
Prefix discovery	The process by which hosts discover the network prefixes for local link destinations. Similar to the ICMPv4 Address Mask Request/Reply. For more information, see "Router Discovery."
Parameter discovery	The process by which hosts discover additional operating parameters, including the link MTU and the default hop limit for outgoing packets. For more information, see "Router Discovery."
Address autoconfiguration	The process for configuring IP addresses for interfaces in either the presence or absence of a stateful address configuration server such as Dynamic Host Configuration Protocol version 6 (DHCPv6). For more information, see "Address Autoconfiguration."
Address resolution	The process by which nodes resolve a neighbor's IPv6 address to its link-layer

	address. Equivalent to ARP in IPv4. For more information, see “Address Resolution.”
Next-hop determination	The process by which a node determines the IPv6 address of the neighbor to which a packet is being forwarded based on the destination address. The forwarding or next-hop address is either the destination address or the address of an on-link default router. For more information, see “Sending Host Algorithm.”
Neighbor unreachability detection	The process by which a node determines that the IPv6 layer of a neighbor is no longer receiving packets. For more information, see “Neighbor Unreachability Detection.”
Duplicate address detection	The process by which a node determines that an address considered for use is not already in use by a neighboring node. Equivalent to using gratuitous ARP frames in IPv4. For more information, see “Duplicate Address Detection.”
Redirect function	The process of informing a host of a better first-hop IPv6 address to reach a destination. Equivalent to the use of the IPv4 ICMP Redirect message. For more information, see “Redirect Function.”

### Neighbor Discovery Message Format

Like Multicast Listener Discovery (MLD) messages, ND messages use the ICMPv6 message structure and ICMPv6 types 133 through 137. ND messages consist of an ND message header, composed of an ICMPv6 header and ND message-specific data, and zero or more ND options, as shown in Figure 41.



Figure 41 The format of a Neighbor Discovery message

There are five different ND messages:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

ND message options provide additional information, typically indicating MAC addresses, on-link network prefixes, on-link MTU information, and redirection data.

To ensure that ND messages received have originated from a node on the local link, all ND messages are sent with a hop limit of 255. When an ND message is received, the Hop Limit field in the IPv6 header is checked. If it is not set to 255, the message is silently discarded. Verifying that the ND message has a hop limit of 255 provides protection from ND-based network attacks launched from off-link nodes. With a hop limit of 255, a router could not have forwarded the ND message from an off-link node.

## Neighbor Discovery Options

ND options are formatted in Type-Length-Value format, as shown in Figure 42.

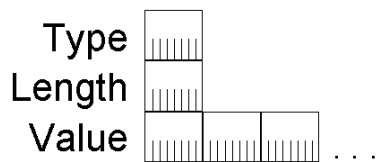


Figure 42 The format of a Neighbor Discovery option

The 8-bit Type field indicates the type of ND option. Table 9 lists the ND option types defined in RFC 4861.

Table 9 IPv6 Neighbor Discovery Option Types

Type	Option Name
1	Source Link-Layer Address
2	Target Link-Layer Address
3	Prefix Information
4	Redirected Header
5	MTU

The 8-bit Length field indicates the length of the entire option in 8-byte blocks. All ND options must fall on 8-byte boundaries. The variable length Value field contains the data for the option.

### Source/Target Link-Layer Address Option

The Source Link-Layer Address option indicates the link-layer address of the ND message sender. The Source Link-Layer Address option is included in the Neighbor Solicitation, Router Solicitation, and Router Advertisement messages. The Source Link-Layer Address option is not included when the source address of the ND message is the unspecified address (::).

The Target Link-Layer Address option indicates the link-layer address of the neighboring node to which IPv6 packets should be directed. The Target Link-Layer Address option is included in the Neighbor Advertisement and Redirect messages.

The Source Link-Layer Address option and the Target Link-Layer Address option have the same format shown in Figure 43.

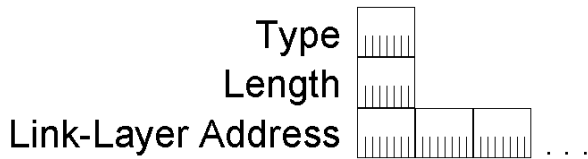


Figure 43 The format of the Source and Target Link-Layer Address options

The Type field is set to 1 for a Source Link-Layer Address option and 2 for a Target Link-Layer Address option. The Length field is set to the number of 8-byte blocks in the entire option. The Link-Layer Address field is a variable-length field that contains the link-layer address of the source or target. Each link layer defined for IPv6 must specify the way in which the link-layer address is formatted in the Source and Target Link-Layer Address options.

For example, RFC 2464 defines how IPv6 packets are sent over Ethernet networks. It also includes the format of the Source and Target Link-Layer Address ND options. For Ethernet, the link-layer address is 48-bits (6-bytes) in length. Figure 44 shows the Source and Target Link-Layer Address options for Ethernet.

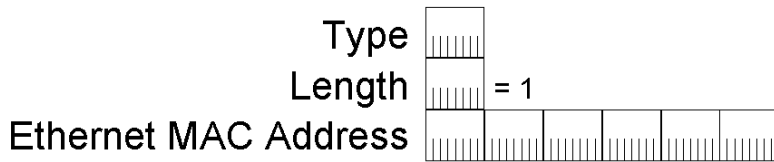


Figure 44 The format of the Source and Target Link-Layer Address options for Ethernet

### Prefix Information Option

The Prefix Information option is sent in Router Advertisement messages to indicate both address prefixes and information about address autoconfiguration. There can be multiple Prefix Information options included in a Router Advertisement message, indicating multiple address prefixes. Figure 45 shows the format of the Prefix Information option.

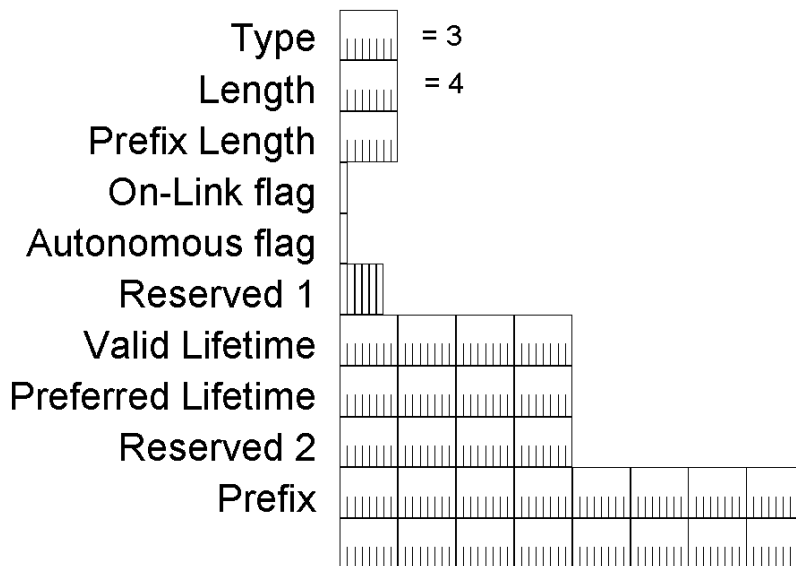


Figure 45 The format of the Prefix Information option

The fields in the Prefix Information option are:

- **Type** – The value of this field is 3.
- **Length** – The value of this field is 4 (the entire option is 32 bytes in length).
- **Prefix Length** – Indicates the number of leading bits in the Prefix field that comprise the address prefix. The size of this field is 8 bits. The Prefix Length field has a value from 0 to 128.
- **On-link flag** – Indicates, when set to 1, that the addresses implied by the included prefix are available on the link on which this Router Advertisement message was received. When set to 0, it is not assumed that the addresses that match the prefix are available on-link. The size of this field is 1 bit.
- **Autonomous flag** – Indicates, when set to 1, that the included prefix is used to create an autonomous (or stateless) address configuration. When set to 0, the included prefix is not used to create a stateless address configuration. The size of this field is 1 bit.
- **Reserved 1** – A 6-bit field reserved for future use and set to 0.
- **Valid Lifetime** – Indicates the number of seconds that an address, based on the included prefix and using stateless address configuration, remains valid. The size of this field is 32 bits. The Valid Lifetime field also indicates the number of seconds that the included prefix is valid for on-link determination. For an infinite valid lifetime, the Valid Lifetime field is set to 0xFFFFFFFF.
- **Preferred Lifetime** – Indicates the number of seconds that an address, based on the included prefix and using stateless address configuration, remains in a preferred state. The size of this field is 32 bits. Stateless autoconfiguration addresses that are still valid are either in a preferred or deprecated state. In the preferred state, the address can be used for unrestricted communication. In the deprecated state, the use of the address is not recommended for new communications. However, existing communications using a deprecated address can continue. An address goes from the preferred state to the deprecated state when its preferred lifetime expires. For an infinite preferred lifetime, the Preferred Lifetime field is set to 0xFFFFFFFF.
- **Reserved 2** – A 32-bit field reserved for future use and set to 0.
- **Prefix** – Indicates the prefix for the IPv6 address derived through stateless autoconfiguration. The size of this field is 128 bits. The combination of the Prefix Length field and the Prefix field unambiguously describe the prefix which, when combined with the interface identifier for the node, create an IPv6 address. Bits in the Prefix field that are within the length of the Prefix Length field are significant. The link-local prefix should not be sent and is ignored by the receiving host.

### Redirected Header Option

The Redirected Header option is sent in Redirect messages to specify the IPv6 packet that caused the router to send a Redirect message. It can contain all or part of the redirected IPv6 packet, depending on the size of the IPv6 packet that was initially sent. Figure 46 shows the format of the Redirected Header option.

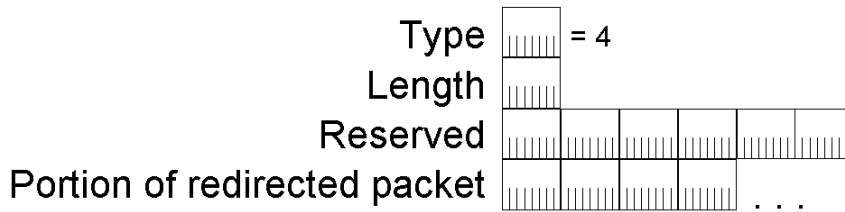


Figure 46 The format of the Redirected Header option

The fields in the Redirected Header option are:

- **Type** – The value of this field is 4.
- **Length** – The value of this field is the number of 8-byte blocks in the entire option.
- **Reserved** – A 48-bit field reserved for future use and set to 0.
- **Portion of redirected packet** – Contains either the IPv6 packet or a portion of the IPv6 packet that caused the Redirect message to be sent. The amount of the original packet that is included is the portion of the packet so that the entire Redirect message is no more than 1280 bytes in length.

### MTU Option

The MTU option is sent in Router Advertisement messages to indicate the IPv6 MTU of the link. This option is typically used only when the IPv6 MTU for a link is not well known or needs to be set due to a translational or mixed-media bridging configuration. The MTU option overrides the IPv6 MTU reported by the interface hardware.

In bridged or Layer-2 switched environments, it is possible to have different link-layer technologies with different link-layer MTUs on the same network segment. In this case, differences in IPv6 MTUs between nodes on the same network are not discovered through Path MTU Discovery. The MTU option is used to indicate the highest IPv6 MTU supported by all link-layer technologies on the network segment.

Consider the switched configuration shown in Figure 47.

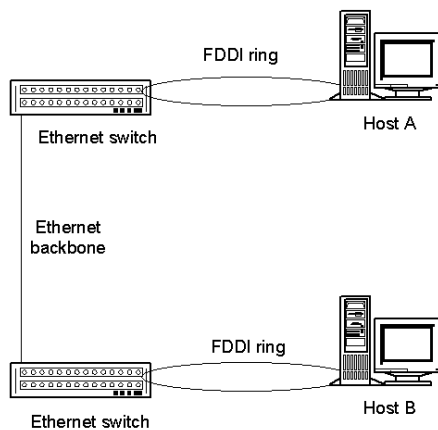


Figure 47 A Layer 2 switched environment that is utilizing the MTU option

Two IPv6 hosts, Host A and Host B, are connected to two different Ethernet (Layer 2) switches using Fiber Distributed Data Interface (FDDI) ports. The two switches are connected by an Ethernet backbone. When Host A and Host B negotiate a TCP connection, each reports a TCP maximum segment size of 4312 (the FDDI link-layer MTU of 4352 minus 40 bytes of IPv6 header). When TCP data on the connection begins to flow, the switches silently discard IPv6 packets larger than 1500 bytes that are sent between Host A and Host B.

With the MTU option, the router for the network segment (not shown) reports an IPv6 MTU of 1500 in the Router Advertisement message for all hosts on the network segment. When both Host A and Host B adjust their IPv6 MTU from 4312 to 1500, maximum-sized TCP connection data between them is not discarded by the intermediate switches.

The format of the MTU option is shown in Figure 48.

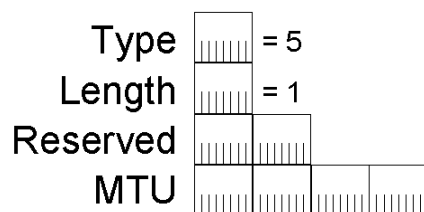


Figure 48 The format of the MTU option

The fields in the MTU option are:

- **Type** – The value of this field is 5.
- **Length** – The value of this field is 1 (there are 8 bytes in the entire option).
- **Reserved** – A 16-bit field reserved for future use and set to 0.
- **MTU** – Indicates the IPv6 MTU that should be used by the host for the link on which the Router Advertisement was received. The size of this field is 32 bits. The value in the MTU field is ignored if it is larger than the link MTU.

## Neighbor Discovery Messages

All of the functions of IPv6 ND are performed with the following messages:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

### Router Solicitation

The Router Solicitation message is sent by IPv6 hosts to discover IPv6 routers present on the link. A host sends a multicast Router Solicitation to prompt IPv6 routers to respond immediately, rather than waiting for a periodic Router Advertisement message.

For example, assuming the local link is Ethernet, in the Ethernet header of the Router Solicitation message:

- The Source Address field is set to the MAC address of the sending network adapter.
- The Destination Address field is set to 33-33-00-00-00-02.

In the IPv6 header of the Router Solicitation message:

- The Source Address field is set to either a link-local IPv6 address assigned to the sending interface or the IPv6 unspecified address (::).
- The Destination Address field is set to the link-local scope all-routers multicast address (FF02::2).
- The Hop Limit field is set to 255.

The format of the Router Solicitation message is shown in Figure 49.

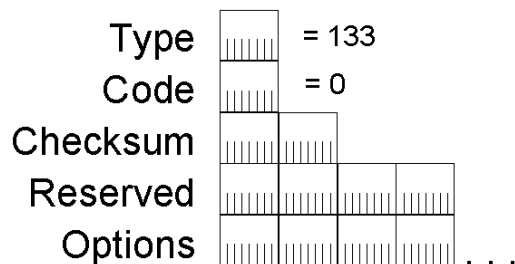


Figure 49 The format of the Router Solicitation message

The fields in the Router Solicitation message are:

- **Type** – The value of this field is 133.
- **Code** – The value of this field is 0.
- **Checksum** – The value of this field is the ICMPv6 checksum.
- **Reserved** – A 32-bit field reserved for future use and set to 0.
- **Source Link-Layer Address option** – The ND Source Link-Layer Address option contains the link-layer address of the sender. For an Ethernet node, the Source Link-Layer Address option contains the Ethernet MAC address of the sending host. The address in the Source Link-Layer Address option is used by the receiving router to determine the unicast MAC address of the host to which the corresponding unicast Router Advertisement is sent.

## Router Advertisement

IPv6 routers send the Router Advertisement message either periodically or in response to the receipt of a Router Solicitation message. It contains the information required by hosts to determine the link prefixes, the link MTU, whether or not to use address autoconfiguration, and the duration for which addresses created through address autoconfiguration are both valid and preferred.

For example, assuming the local link is Ethernet, in the Ethernet header of the Router Advertisement message:

- The Source Address field is set to the MAC address of the sending network adapter.

- The Destination Address field is set to either 33-33-00-00-00-01 for a periodic Router Advertisement or the unicast MAC address of the host that sent a Router Solicitation.

In the IPv6 header of the Router Advertisement message:

- The Source Address field is set to the link-local address assigned to the sending interface.
- The Destination Address field is set to either the link-local scope all-nodes multicast address (FF02::1) or the unicast IPv6 address of the host that sent the Router Solicitation message.
- The Hop Limit field is set to 255.

The format of the Router Advertisement message is shown in Figure 50.

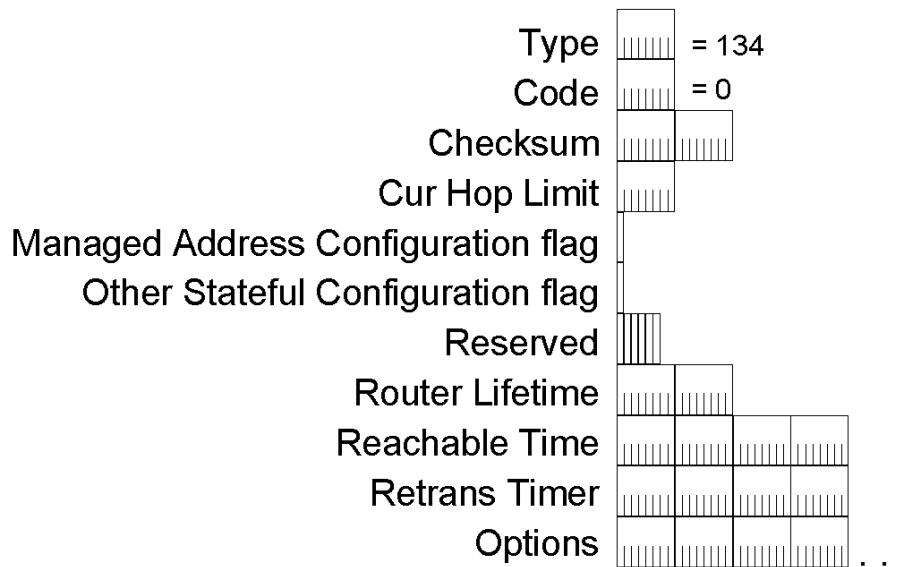


Figure 50 The format of the Router Advertisement message

The fields in the Router Advertisement message are:

- **Type** – The value of this field is 134.
- **Code** – The value of this field is 0.
- **Checksum** – The value of this field is the ICMPv6 checksum.
- **Cur Hop Limit** – Indicates the default value of the Hop Count field in the IPv6 header for packets sent by hosts that receive this Router Advertisement message. The size of this field is 8 bits. A Cur Hop Limit of 0 indicates that the default value of the Hop Count field is not specified by the router.
- **Managed Address Configuration flag** – Indicates, when set to 1, that hosts receiving this Router Advertisement message must use a stateful address configuration protocol (for example, DHCPv6) to obtain addresses in addition to the addresses derived from stateless address autoconfiguration. The size of this field is 1 bit.
- **Other Stateful Configuration flag** – Indicates, when set to 1, that hosts receiving this Router Advertisement message must use a stateful address configuration protocol (for example, DHCPv6) to obtain non-address configuration information. The size of this field is 1 bit.

- **Reserved** – A 6-bit field reserved for future use and set to 0.
- **Router Lifetime** – Indicates the lifetime (in seconds) of the router as the default. The size of this field is 16 bits. The maximum Router Lifetime value is 65,535 seconds (about 18.2 hours). A Router Lifetime of 0 indicates that the router cannot be considered a default router. All other information contained in the Router Advertisement, however, is valid.
- **Reachable Time** – Indicates the amount of time (in milliseconds) that a node can consider a neighboring node reachable after receiving a reachability confirmation. The size of this field is 32 bits. A Reachable Time value of 0 indicates that the router does not specify the Reachable Time. For more information, see “Neighbor Reachability Detection.”
- **Retrans Timer** – Indicates the amount of time (in milliseconds) between retransmissions of Neighbor Solicitation messages. The size of this field is 32 bits. Retrans Timer is used during Neighbor Unreachability Detection. A Retrans Timer value of 0 indicates that the router does not specify the Retrans Timer.
- **Source Link-Layer Address option** – The Source Link-Layer Address option contains the link-layer address of the interface on which the Router Solicitation message was sent. This option can be omitted when the router is load balancing across multiple link-layer addresses.
- **MTU option** – The MTU option contains the MTU of the link. It should be sent only on links that have a variable MTU or in switched environments with multiple link-layer technologies on the same network segment.
- **Prefix Information options** – The prefix information options contain the on-link prefixes that are used for address autoconfiguration. The local-link prefix is never sent as a prefix information option.

### Neighbor Solicitation

The Neighbor Solicitation message is sent by IPv6 hosts to discover the link-layer address of an on-link IPv6 node. It includes the link-layer address of the sender. Typical Neighbor Solicitations are multicast for address resolution and unicast when the reachability of a neighboring node is being verified.

For example, assuming the local link is Ethernet, in the Ethernet header of the Neighbor Solicitation message:

- The Source Address field is set to the MAC address of the sending network adapter.
- For a multicast Neighbor Solicitation, the Destination Address field is set to the Ethernet MAC address that corresponds to the solicited-node multicast address of the target. For a unicast Neighbor Solicitation, the Destination Address field is set to the unicast MAC address of the neighbor.

In the IPv6 header of the Neighbor Solicitation message:

- The Source Address field is set to either an IPv6 address assigned to the sending interface or, during duplicate address detection, the unspecified address (::).
- For a multicast Neighbor Solicitation, the Destination Address field is set to the solicited-node multicast address of the target. For a unicast Neighbor Solicitation, the Destination Address field is set to the unicast or anycast address of the target.
- The Hop Limit field is set to 255.

The format of the Neighbor Solicitation message is shown in Figure 51.

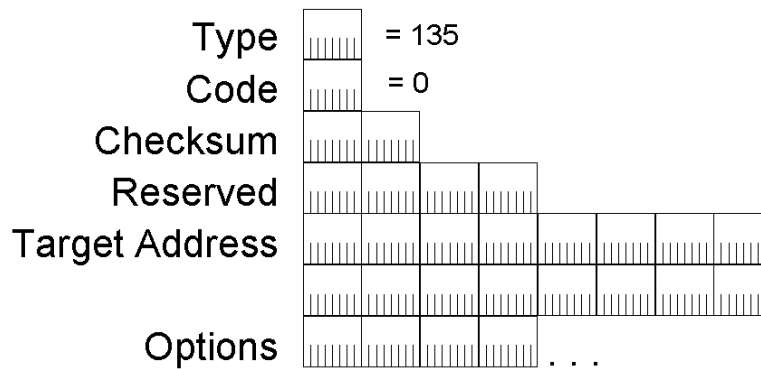


Figure 51 The format of the Neighbor Solicitation message

The fields in the Neighbor Solicitation message are:

- **Type** – The value of this field is 135.
- **Code** – The value of this field is 0.
- **Checksum** – The value of this field is the ICMPv6 checksum.
- **Reserved** – A 32-bit field reserved for future use and set to 0.
- **Target Address** – Indicates the IP address of the target. The size of this field is 128 bits.
- **Source Link-Layer Address option** – The Source Link-Layer Address option contains the link-layer address of the sender. For an Ethernet node, the Source Link-Layer Address option contains the Ethernet MAC address of the sending node. The address in the Source Link-Layer Address option is used by the receiving node to determine the unicast MAC address of the node to which the corresponding Neighbor Advertisement is sent. During duplicate address detection, when the source IPv6 address is the unspecified address (::), the Source Link-Layer Address option is not included.

### Neighbor Advertisement

The Neighbor Advertisement message is sent by an IPv6 node in response to the receipt of a Neighbor Solicitation message. An IPv6 node also sends unsolicited Neighbor Advertisements to inform neighboring nodes of changes in link-layer addresses. The Neighbor Advertisement contains information required by nodes to determine the type of Neighbor Advertisement message, the link-layer address of the sender, and the sender's role on the network.

For example, assuming the local link is Ethernet, in the Ethernet header of the Neighbor Advertisement message:

- The Source Address field is set to the MAC address of the sending network adapter.
- The Destination Address field is set, for a solicited Neighbor Advertisement, to the unicast MAC address of the initial Neighbor Solicitation sender. For an unsolicited Neighbor Advertisement, the Destination Address field is set to 33-33-00-00-00-01, the Ethernet MAC address corresponding to the link-local scope all-nodes multicast address.

In the IPv6 header of the Neighbor Advertisement message:

- The Source Address field is set to the link-local address assigned to the sending interface.

- The Destination Address field is set, for a solicited Neighbor Advertisement, to the unicast IP address of the sender of the initial Neighbor Solicitation. For an unsolicited Neighbor Advertisement, the Destination Address field is set to the link-local scope all-nodes multicast address (FF02::1).
- The Hop Limit field is set to 255.

The format of the Neighbor Advertisement message is shown in Figure 52.

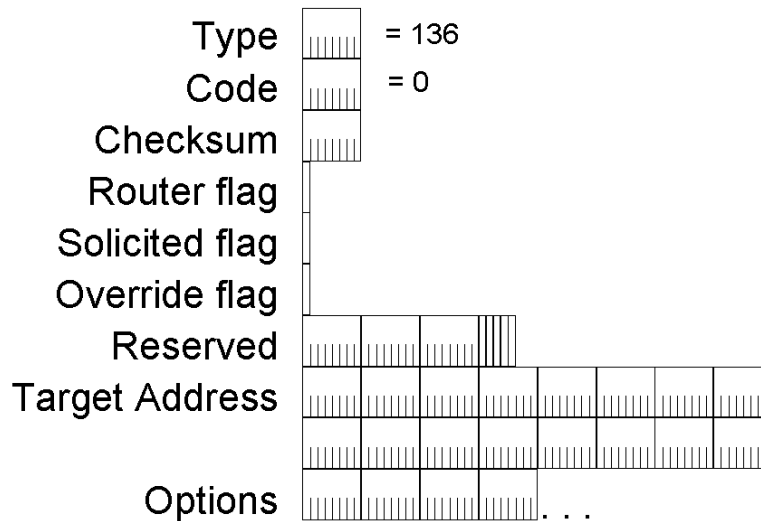


Figure 52 The format of the Neighbor Advertisement message

The fields in the Neighbor Advertisement message are:

- **Type** – The value of this field is 136.
- **Code** – The value of this field is 0.
- **Checksum** – The value of this field is the ICMPv6 checksum.
- **Router flag** – Indicates the role of the sender of the Neighbor Advertisement message. The size of this field is 1 bit. The Router flag is set to 1 when the sender is a router and 0 when the sender is not. The Router flag is used by Neighbor Unreachability Detection to determine when a router changes to a host.
- **Solicited flag** – Indicates, when set to 1, that the Neighbor Advertisement message was sent in response to a Neighbor Solicitation message. The size of this field is 1 bit. The Solicited flag is used as a reachability confirmation during Neighbor Unreachability Detection. The Solicited flag is set to 0 for both multicast Neighbor Advertisements and unsolicited unicast Neighbor Advertisements.
- **Override flag** – Indicates, when set to 1, that the link-layer address in the included Target Link-Layer Address option should override the link-layer address in the existing neighbor cache entry. The size of this field is 1 bit. If the Override flag is set to 0, the enclosed link-layer address only updates a neighbor cache entry if the link-layer address is not known. The Override flag is set to 0 for solicited anycast address and proxied advertisements. The Override flag is set to 1 in other solicited and unsolicited advertisements. For more information on the neighbor cache, see “Neighbor Discovery Processes.”
- **Reserved** – A 29-bit field reserved for future use and set to 0.

- **Target Address** – Indicates the address being advertised. The size of this field is 128 bits. For solicited Neighbor Advertisement messages, the target address is contained in the Target Address field in the corresponding Neighbor Solicitation. For unsolicited Neighbor Advertisement messages, the target address is the address whose link-layer address has changed.
- **Target Link-Layer Address option** – The Target Link-Layer Address option contains the link-layer address of the target, which is the sender of the Neighbor Advertisement. For an Ethernet node, the Target Link-Layer Address option contains the Ethernet MAC address of sending node. The address in the Target Link-Layer Address option is used by receiving nodes to determine the unicast MAC address of the advertising node.

**Redirect**

The Redirect message is sent by an IPv6 router to inform an originating host of a better first-hop address for a specific destination. Redirect messages are only sent by routers for unicast traffic, are only unicast to originating hosts, and are only processed by hosts.

For example, assuming the local link is Ethernet, in the Ethernet header of the Redirect message:

- The Source Address field is set to the MAC address of the sending network adapter.
- The Destination Address field is set to the unicast MAC address of the originating sender.

In the IPv6 header of the Redirect message:

- The Source Address field is set to the link-local address that is assigned to the sending interface.
- The Destination Address field is set to the unicast IP address of the originating host.
- The Hop Limit field is set to 255.

The format of the Redirect message is shown in Figure 53.

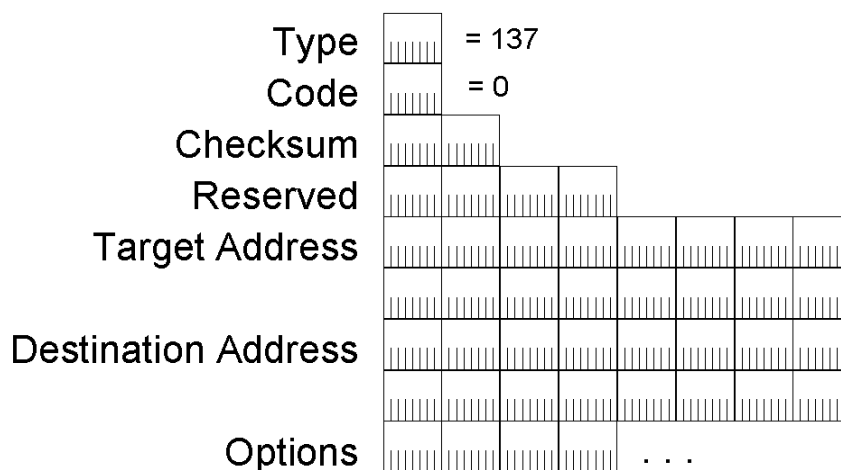


Figure 53 The format of the Redirect message

The fields in the Redirect message are:

- **Type** – The value of this field is 137.
- **Code** – The value of this field is 0.

- **Checksum** – The value of this field is the ICMPv6 checksum.
- **Reserved** – A 32-bit field reserved for the future and set to 0.
- **Target Address** – Indicates the better next-hop address for packets addressed to the node in the Destination Address field. The size of this field is 128 bits. For off-link traffic, the Target Address field is set to the local-link address of a local router. For on-link traffic, the Target Address field is set to the Destination Address field in the Redirect message.
- **Destination Address** – Contains the Destination Address of the packet that caused the router to send the Redirect message. The size of this field is 128 bits. Upon receipt at the originating host, the Target Address and Destination Address fields are used to update forwarding information for the destination. Subsequent packets sent to the destination by the host are forwarded to the address in the Target Address field.
- **Target Link-Layer Address option** – The Target Link-Layer Address option contains the link-layer address of the target (the node to which subsequent packets should be sent.) The Target Link-Layer Address option can be included only when known by the router.
- **Redirected Header option** – The Redirected Header option includes a portion of the original packet that caused the Redirect message to be sent so that the entire IPv6 packet containing the Redirect message is no larger than 1280 bytes.

## Neighbor Discovery Processes

The ND protocol provides message exchanges for the following processes:

- Address resolution (including duplicate address detection)
- Router discovery (includes prefix and parameter discovery)
- Neighbor unreachability detection
- Redirect function

For information on address autoconfiguration, see “Address Autoconfiguration.” For information on next-hop determination, see “Sending Host Algorithm.”

To facilitate interactions between neighboring nodes, RFC 4861 defines the following host data structures as an example of how to store information for ND processes:

- **Neighbor cache**  
Stores the on-link IP address of a neighbor, its corresponding link-layer address, and an indication of the neighbor’s reachability state. The neighbor cache is equivalent to the ARP cache in IPv4.
- **Destination cache**  
Stores information on forwarding or next-hop IP addresses for destinations to which traffic has recently been sent. Entries in the destination cache contain the destination IP address (either local or remote), the previously resolved next-hop IP address, and the Path MTU for the destination.
- **Prefix list**

Lists on-link prefixes. Each entry in the prefix list defines a range of IP addresses for destinations that are directly reachable (neighbors). This list is populated from prefixes advertised by routers in the Router Advertisement message.

- **Default router list**

Lists IP addresses corresponding to on-link routers that send Router Advertisement messages and are eligible to be default routers.

RFC 4861 defines these data structures as an example of an IPv6 host conceptual model. An IPv6 implementation is not required to create these exact data structures as long as the external behavior of the host is consistent with RFC 4861. For example, all Microsoft IPv6 implementations use a routing table rather than a prefix list and default router list.

### **Address Resolution**

The address resolution process for IPv6 nodes consists of an exchange of Neighbor Solicitation and Neighbor Advertisement messages to resolve the link-layer address of the on-link next-hop address for a given destination. The sending host sends a multicast Neighbor Solicitation message on the appropriate interface. The multicast address of the Neighbor Solicitation message is the solicited-node multicast address derived from the target IP address. The Neighbor Solicitation message includes the link-layer address of the sending host in the Source Link-Layer Address option. For information on how a host determines the next-hop address for a destination, see “Host Sending Algorithm.”

When the target host receives the Neighbor Solicitation message, it updates its own neighbor cache based on the source address of the Neighbor Solicitation message and the link-layer address in the Source Link-Layer Address option. Next, the target node sends a unicast Neighbor Advertisement to the Neighbor Solicitation sender. The Neighbor Advertisement includes the Target Link-Layer Address option.

After receiving the Neighbor Advertisement from the target, the sending host updates its neighbor cache with an entry for the target based upon the information in the Target Link-Layer Address option. At this point, unicast IPv6 traffic between the sending host and the target of the Neighbor Solicitation can be sent.

#### *Address Resolution Example*

Host A has an Ethernet MAC address of 00-AA-00-11-11-11 and a corresponding link-local address of FE80::2AA:FF:FE11:1111. Host B has an Ethernet MAC address of 00-AA-00-22-22-22 and a corresponding link-local address of FE80::2AA:FF:FE22:2222. To send a packet to Host B, Host A must use address resolution to resolve Host B's link-layer address.

Based on Host B's IP address, Host A sends a solicited-node multicast Neighbor Solicitation to the address of FF02::1:FF22:2222, as shown in Figure 54.

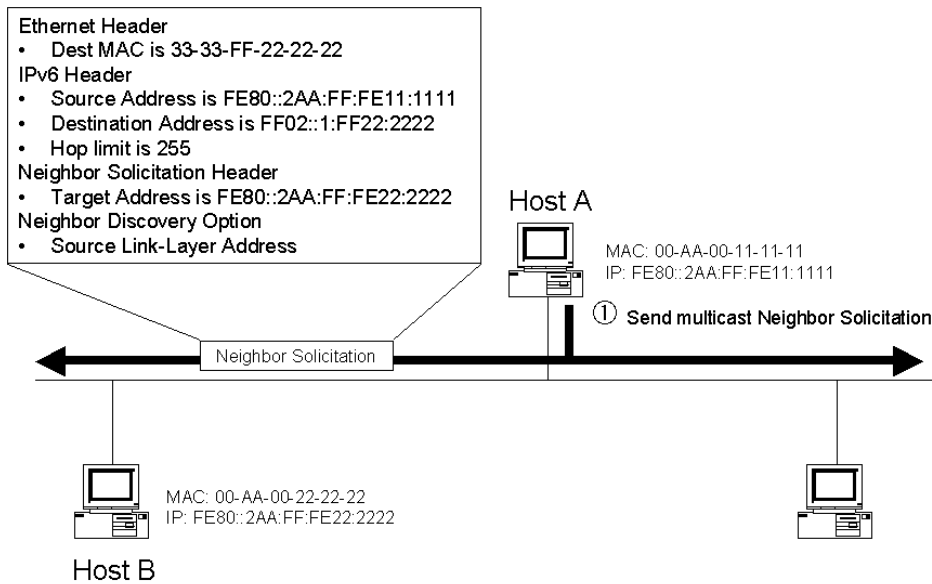


Figure 54 The multicast Neighbor Solicitation for address resolution

Host B, having registered the solicited-node multicast address of 33-33-FF-22-22-22 with its Ethernet adapter, receives and processes the Neighbor Solicitation. Host B responds with a unicast Neighbor Advertisement message, as shown in Figure 55.

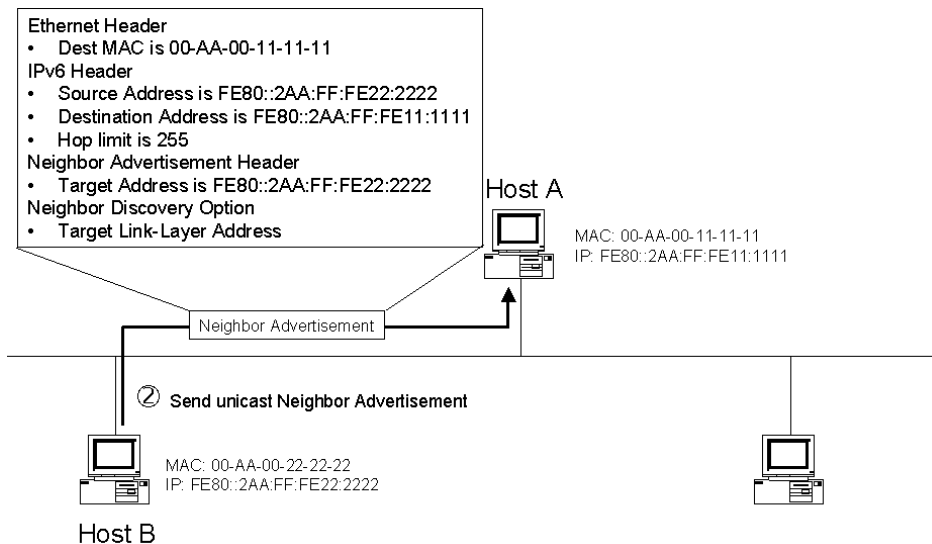


Figure 55 The unicast Neighbor Advertisement for address resolution

### Duplicate Address Detection

IPv4 nodes use ARP Request messages and a method called gratuitous ARP to detect a duplicate IP address on the local link. Similarly, IPv6 nodes use the Neighbor Solicitation message to detect duplicate address use on the local link.

With IPv4 gratuitous ARP, the Source Protocol Address and Target Protocol Address fields in the ARP Request message header are set to the IPv4 address for which duplication is being detected. In IPv6

duplicate address detection, the Target Address field in the Neighbor Solicitation message is set to the IPv6 address for which duplication is being detected.

Duplicate address detection differs from address resolution in these ways:

- In the duplicate address detection Neighbor Solicitation message, the Source Address field in the IPv6 header is set to the unspecified address (::). The address being queried for duplication cannot be used until it is determined that there are no duplicates.
- In the Neighbor Advertisement reply to a duplicate address detection Neighbor Solicitation message, the Destination Address in the IP header is set to the link-local scope all-nodes multicast address (FF02::1). The Solicited flag in the Neighbor Advertisement message is set to 0. Because the sender of the duplicate address detection Neighbor Solicitation message is not using the desired IP address, it cannot receive unicast Neighbor Advertisements. Therefore, the Neighbor Advertisement is multicast.

Upon receipt of the multicast Neighbor Advertisement with the Target Address field set to the IP address for which duplication is being detected, the node disables the use of the duplicate IP address on the interface. If the node does not receive a Neighbor Advertisement that defends the use of the IPv6 address, it initializes the address on the interface.

#### *Duplicate Address Detection Example*

Host B has a link-local address of FE80::2AA:FF:FE22:2222. Host A is attempting to use the link-local address of FE80::2AA:FF:FE22:2222. However, before Host A can use this link-local address, it must verify its uniqueness through duplicate address detection.

Host A sends a solicited-node multicast Neighbor Solicitation to the address FF02::1:FF22:2222, as shown in Figure 56.

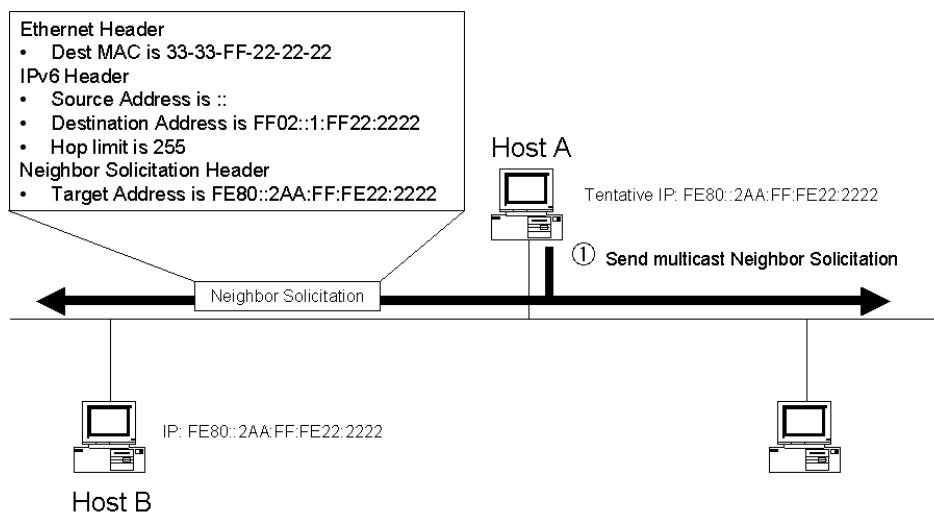


Figure 56 The multicast Neighbor Solicitation for duplicate address detection

Host B, having registered the solicited-node multicast address of 33-33-FF-22-22-22 with its Ethernet adapter, receives and processes the Neighbor Solicitation. Host B notes that the source address is the unspecified address. Host B then responds with a multicast Neighbor Advertisement message, as shown in Figure 57.

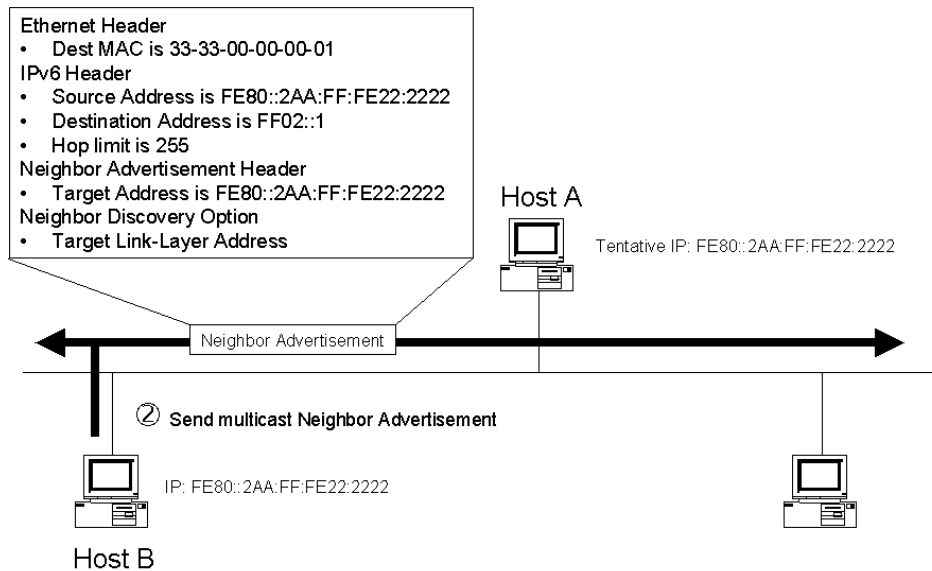


Figure 57 The multicast Neighbor Advertisement for duplicate address detection

## Router Discovery

Router discovery is the process through which nodes attempt to discover the set of routers on the local link. Router discovery in IPv6 is similar to ICMP Router Discovery for IPv4.

An important difference between ICMPv4 Router Discovery and IPv6 Router Discovery is the mechanism through which a new default router is selected when the current one becomes unavailable. In ICMPv4 Router Discovery, the Router Advertisement message includes an Advertisement Lifetime field. Advertisement Lifetime is the time after which the router, upon receiving its last Router Advertisement message, can be considered unavailable. In the worst case, a router can become unavailable and hosts will not attempt to discover a new default router until the Router Advertisement time has elapsed.

IPv6 has a Router Lifetime field in the Router Advertisement message. This field indicates the length of time that the router can be considered a default router. However, if the current default router becomes unavailable, the condition is detected through neighbor unreachability detection instead of the Router Lifetime field in the Router Advertisement message. Because neighbor unreachability detection determines that the router is no longer reachable, a new router is chosen immediately from the default router list. For more information, see “Neighbor Unreachability Detection.”

In addition to configuring a default router, IPv6 router discovery also configures the following:

- The default setting for the Hop Limit field in the IPv6 header.
- A determination of whether the node should use a stateful address protocol, such as Dynamic Host Configuration Protocol for IPv6 (DHCPv6), for addresses and other configuration parameters.
- The timers used in reachability detection and the retransmission of Neighbor Solicitations.
- The list of network prefixes defined for the link. Each network prefix contains both the IPv6 network prefix and its valid and preferred lifetimes. If indicated, a network prefix combined with the interface identifier creates a stateless IP address configuration for the receiving interface. A network prefix also defines the range of addresses for nodes on the local link.

- The MTU of the local link.

The IPv6 router discovery processes are the following:

- IPv6 routers periodically send a Router Advertisement message on the local link advertising their existence as routers. They also provide configuration parameters such as default hop limit, MTU, and prefixes.
- Active IPv6 hosts on the local link receive the Router Advertisement messages and use the contents to maintain the default router list, the prefix list, and other configuration parameters.
- A host that is starting up sends a Router Solicitation message to the link-local scope all-routers multicast address (FF02::2). Upon receipt of a Router Solicitation message, all routers on the local link send a unicast Router Advertisement message to the node that sent the Router Solicitation. The node receives the Router Advertisement messages and uses their contents to build the default router and prefix lists and set other configuration parameters. The number of Router Solicitations sent before abandoning the router discovery process is set by a configurable variable. RFC 4861 uses the variable name of MAX\_RTR\_SOLICITATIONS and recommends a value of 3.

#### *Router and Prefix Discovery Example*

Host A has the Ethernet MAC address of 00-AA-00-11-11-11 and a corresponding link-local address of FE80::2AA:FF:FE11:1111. Router 1 has an Ethernet MAC address of 00-AA-00-22-22-22 and a corresponding link-local address of FE80::2AA:FF:FE22:2222. To forward packets to off-link destinations, Host A must discover the presence of Router 1.

Host A sends a multicast Router Solicitation to the address FF02::2, as shown in Figure 58.

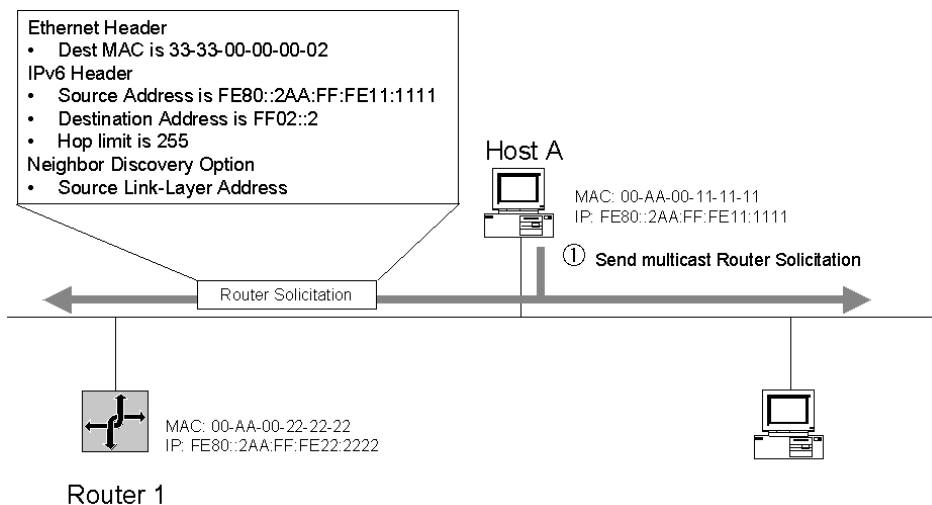


Figure 58 The multicast Router Solicitation for router and prefix discovery

Router 1, having registered the multicast address of 33-33-00-00-00-02 with its Ethernet adapter, receives and processes the Router Solicitation. Router 1 responds with a unicast Router Advertisement message containing configuration parameters and local link prefixes, as shown in Figure 59.

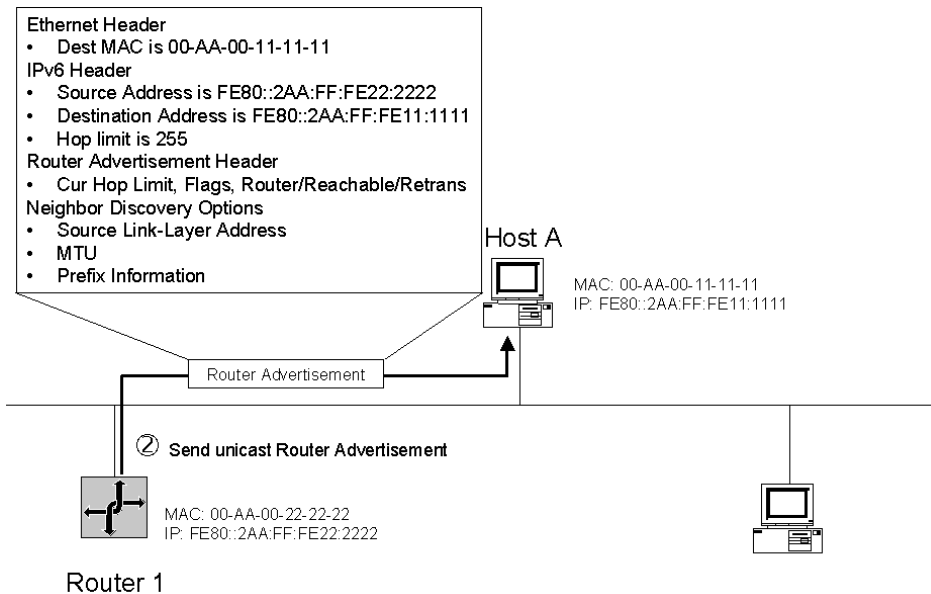


Figure 59 The unicast Router Advertisement for router and prefix discovery

## Neighbor Unreachability Detection

A neighboring node is reachable if there has been a recent confirmation that IPv6 packets sent to the neighboring node were received and processed by the neighboring node. Neighbor unreachability does not necessarily verify the end-to-end reachability of the destination. Because a neighboring node can be a host or router, the neighboring node might not be the final destination of the packet. Neighbor unreachability verifies only the reachability of the first hop to the destination.

One of the ways that reachability is confirmed is through the sending of a unicast Neighbor Solicitation message and the receipt of a solicited Neighbor Advertisement message. A solicited Neighbor Advertisement message, which has its Solicited flag set to 1, is sent only in response to a Neighbor Solicitation message. Unsolicited Neighbor Advertisement or Router Advertisement messages are not considered proof of reachability. The exchange of Neighbor Solicitation and Neighbor Advertisement messages confirms only the reachability of the node that sent the Neighbor Advertisement from the node that sent the Neighbor Solicitation. It does not confirm the reachability of the node that sent the Neighbor Solicitation from the node that sent the Neighbor Advertisement.

For example, if Host A sends a unicast Neighbor Solicitation to Host B and Host B sends a solicited unicast Neighbor Advertisement to Host A, Host A considers Host B reachable. Because there is no confirmation in this exchange that Host A actually received the Neighbor Advertisement, Host B does not consider Host A reachable. To confirm reachability of Host A from Host B, Host B must send its own unicast Neighbor Solicitation to Host A and receive a solicited unicast Neighbor Advertisement from Host A.

Another method of determining reachability is when upper-layer protocols indicate that the communication using the next-hop address is making forward progress. For TCP traffic, forward progress is determined when acknowledgement segments for sent data are received. The end-to-end reachability confirmed by the receipt of TCP acknowledgments implies the reachability of the first hop to the destination. The TCP module provides these indications to the IPv6 protocol module on an ongoing basis.

Other protocols, such as UDP, might not have a method of determining or indicating the forward progress of communication. In this case, the exchange of Neighbor Solicitation and Neighbor Advertisement messages is used to confirm reachability.

The reachability of a neighboring node is determined by monitoring the state of the neighboring node's entry in the neighbor cache. RFC 4861 defines the following states for a neighbor cache entry:

- INCOMPLETE

IPv6 address resolution, which is using a solicited-node multicast Neighbor Solicitation, is in progress. The INCOMPLETE state is entered when a new neighbor cache entry is created but does not yet have the node's corresponding link-layer address. The number of multicast Neighbor Solicitations sent before abandoning the address resolution process and removing the neighbor cache entry is set by a configurable variable. RFC 4861 uses the variable name of `MAX_MULTICAST_SOLICIT` and recommends a value of 3.

- REACHABLE

Reachability has been confirmed by receipt of a solicited unicast Neighbor Advertisement. The neighbor cache entry stays in the REACHABLE state until the number of milliseconds indicated in the Reachable Time field in the Router Advertisement elapses. As long as upper layer protocols such as TCP are indicating that communication is making forward progress, the entry stays in the REACHABLE state. Each time an indication of forward progress is made, the reachable time for the entry is refreshed.

- STALE

Reachable time (the duration since the last reachability confirmation was received) has elapsed. The neighbor cache entry goes into the STALE state after the value (milliseconds) in the Reachable Time field in the Router Advertisement message (or a host default value) elapses and remains in this state until a packet is sent to the neighbor. The STALE state is also entered when an unsolicited Neighbor Advertisement that is advertising the link-layer address is received.

- DELAY

To allow time for upper layer protocols to provide reachability confirmation before sending Neighbor Solicitations, the state of the neighbor cache entry enters the DELAY state and waits a configurable period of time after sending a packet. RFC 4861 uses the variable name of `DELAY_FIRST_PROBE_TIME` and recommends a value of 5 seconds. If no reachability confirmation is received by the delay time, then the entry enters the PROBE state and a unicast Neighbor Solicitation is sent.

- PROBE

Reachability confirmation is in progress for a neighbor cache entry that was in the STALE and DELAY states. Unicast Neighbor Solicitation messages are sent at intervals corresponding to the Retrans Timer field in the Router Advertisement message received by this host. The number of Neighbor Solicitations sent before abandoning the reachability detection process and removing the neighbor cache entry is set by a configurable variable. RFC 4861 uses the variable name of `MAX_UNICAST_SOLICITS` and recommends a value of 3.

Figure 60 shows the state diagram of an entry in the neighbor cache.

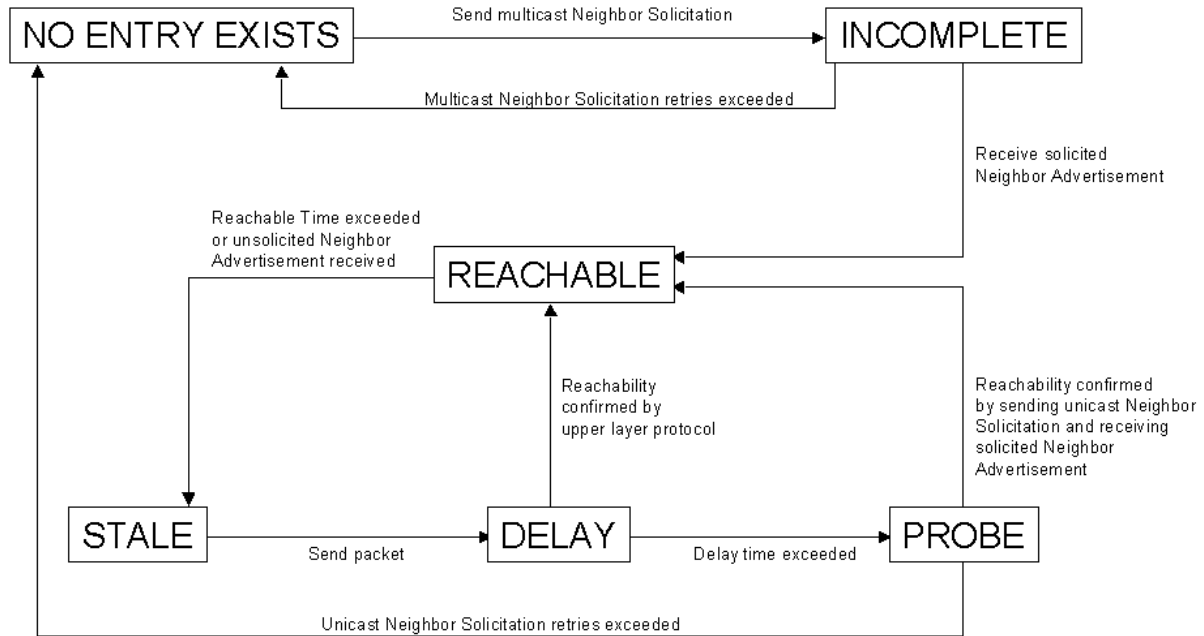


Figure 60 The states of a neighbor cache entry

If the unreachable neighbor is a router, the host chooses another router from the default router list and performs both address resolution and unreachability detection on it.

If a router becomes a host, it should send a multicast Neighbor Advertisement with the Router flag set to 0. If a host receives a Neighbor Advertisement from a router where the Router flag is set to 0, the host removes that router from the default router list and, if necessary, chooses another router.

### Redirect Function

Routers use the redirect function to inform originating hosts of a better first-hop neighbor to which traffic should be forwarded for a specific destination. There are two instances where redirect is used:

1. A router informs an originating host of the IP address of a router available on the local link that is “closer” to the destination. “Closer” is routing metric function used to reach the destination network segment. This condition can occur when there are multiple routers on a network segment and the originating host chooses a default router and it is not the best one to use to reach the destination.
2. A router informs an originating host that the destination is a neighbor (it is on the same link as the originating host). This condition can occur when the prefix list of a host does not include the prefix of the destination. Because the destination does not match a prefix in the list, the originating host forwards the packet to its default router.

The following steps occur in the IPv6 redirect process:

1. The originating host forwards a unicast packet to its default router.
2. The router processes the packet and notes that the address of the originating host is a neighbor. Additionally, it notes that the addresses of both the originating host and the next-hop are on the same link.
3. The router forwards the packet to the appropriate next-hop address.

- The router sends the originating host a Redirect message. In the Target Address field of the Redirect message is the next-hop address of the node to which the originating host should send packets addressed to the destination.

For packets redirected to a router, the Target Address field is set to the link-local address of the router. For packets redirected to a host, the Target Address field is set to the destination address of the packet originally sent.

The Redirect message includes the Redirected Header option. It might also include the Target Link-Layer Address option.

- Upon receipt of the Redirect message, the originating host updates the destination address entry in the destination cache with the address in the Target Address field. If the Target Link-Layer Address option is included in the Redirect message, its contents are used to create or update the corresponding neighbor cache entry.

Redirect messages are only sent by the first router in the path between the originating host and the destination and like ICMPv6 error messages are rate limited. Hosts never send Redirect messages and routers never update routing tables based on the receipt of a Redirect message.

#### Redirect Example

Host A has the Ethernet MAC address of 00-AA-00-11-11-11 and a corresponding link-local address of FE80::2AA:FF:FE11:1111. Host A also has the site-local address of FEC0::1:2AA:FF:FE11:1111. Router 1 has the Ethernet MAC address of 00-AA-00-22-22-22 and a corresponding link-local address of FE80::2AA:FF:FE22:2222. Router 1 also has the site-local address of FEC0::1:2AA:FF:FE22:2222. Router 2 has the Ethernet MAC address of 00-AA-00-33-33-33 and a corresponding link-local address of FE80::2AA:FF:FE33:3333. Router 2 also has the site-local address of FEC0::1:2AA:FF:FE33:3333. Host A is sending a packet to an off-link host at FEC0::2:2AA:FF:FE99:9999 (not shown) and is using Router 1 as its current default router. However, Router 2 is the better router to use to reach this destination.

Host A sends the packet destined to FEC0::2:2AA:FF:FE99:9999 to Router 1, as shown in Figure 61.

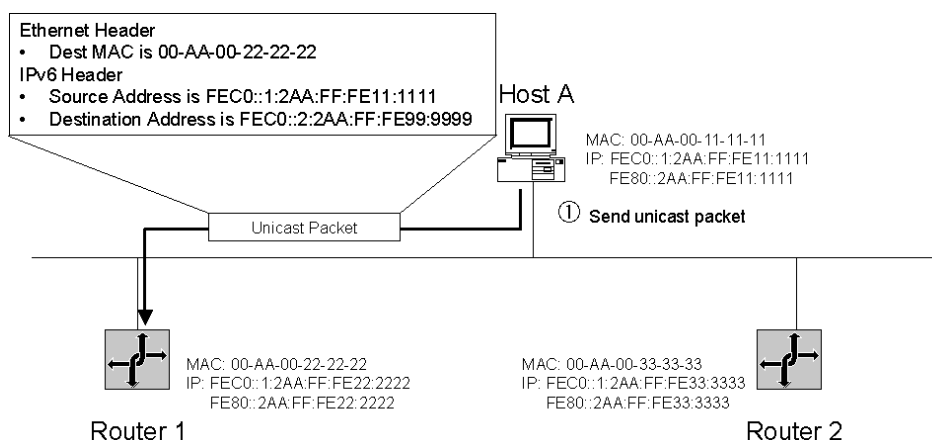


Figure 61 The unicast packet forwarded by the originating node

Router 1 receives the packet from Host A and notes that Host A is a neighbor. It also notes that Host A and the next-hop address for the destination are on the same link. Based on the contents of its local

routing table, Router 1 forwards the unicast packet received from Host A to Router 2, as shown in Figure 62.

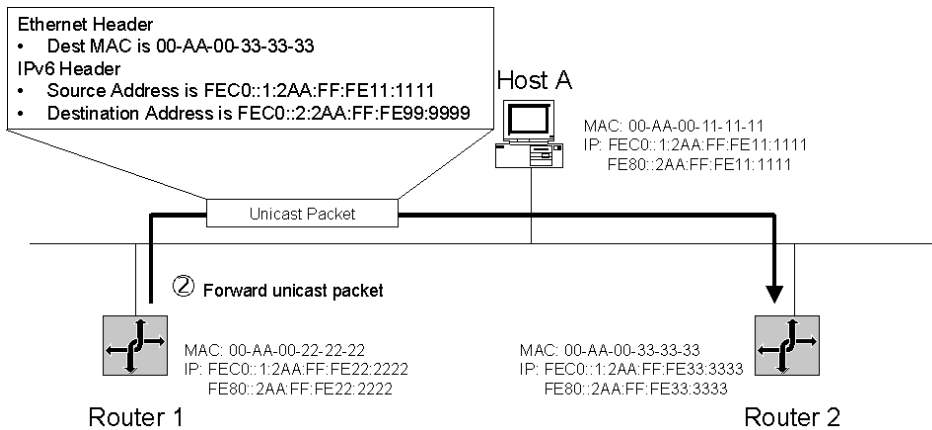


Figure 62 The unicast packet forwarded by the router

To inform Host A that subsequent packets to the destination of FEC0::2:2AA:EE:FE99:9999 should be sent to Router 2, Router 1 sends a Redirect message to Host A, as shown in Figure 63.

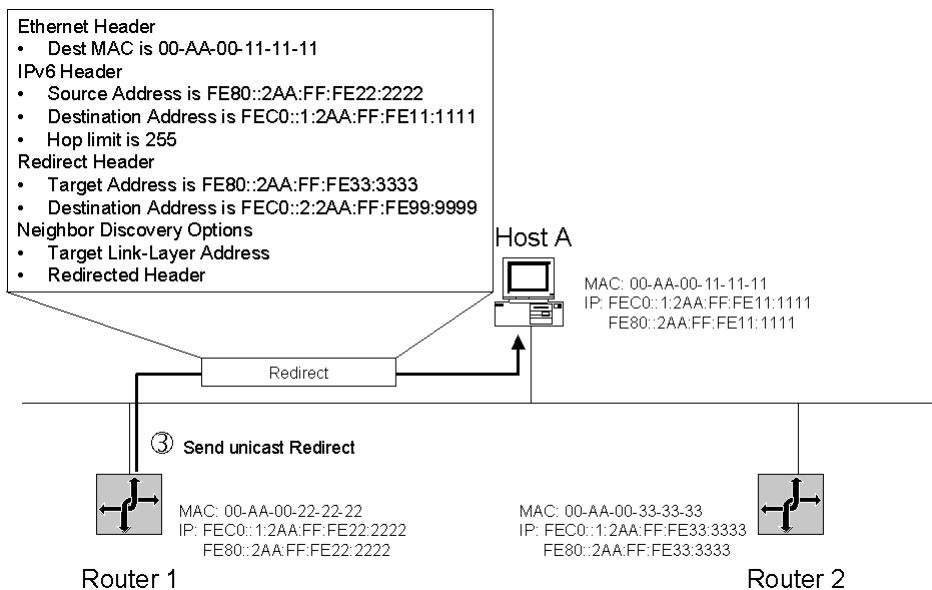


Figure 63 The Redirect message sent by the router

### Host Sending Algorithm

The process by which an IPv6 host sends an IPv6 packet uses a combination of the local host's structures and the ND protocol. An IPv6 host uses the following algorithm when sending a packet to an arbitrary destination:

1. Check the destination cache for an entry matching the destination address.
2. If an entry matching the destination address is found in the destination cache, obtain the next-hop address in the destination cache entry. Go to step 4.

If an entry matching the destination address is not found in the destination cache, determine if the destination address matches a prefix in the prefix list.

If the destination address matches a prefix in the prefix list, the next-hop address is set to the destination address. Go to step 3.

If the destination address does not match a prefix in the prefix list, the next-hop address is set to the address of the current default router. Go to Step 3.

If there is no default router (and there are no routers in the default router list), the next-hop address is set to the destination address.

3. Update the neighbor cache.
4. Check the neighbor cache for an entry matching the next-hop address.
5. If an entry matching the next-hop address is found in the neighbor cache, obtain the link-layer address.

If an entry matching the next-hop address is not found in the neighbor cache, use address resolution to obtain the link-layer address for the next-hop address.

6. Send the packet using the link-layer address of the neighbor cache entry.

The host sending algorithm is shown in Figure 64.

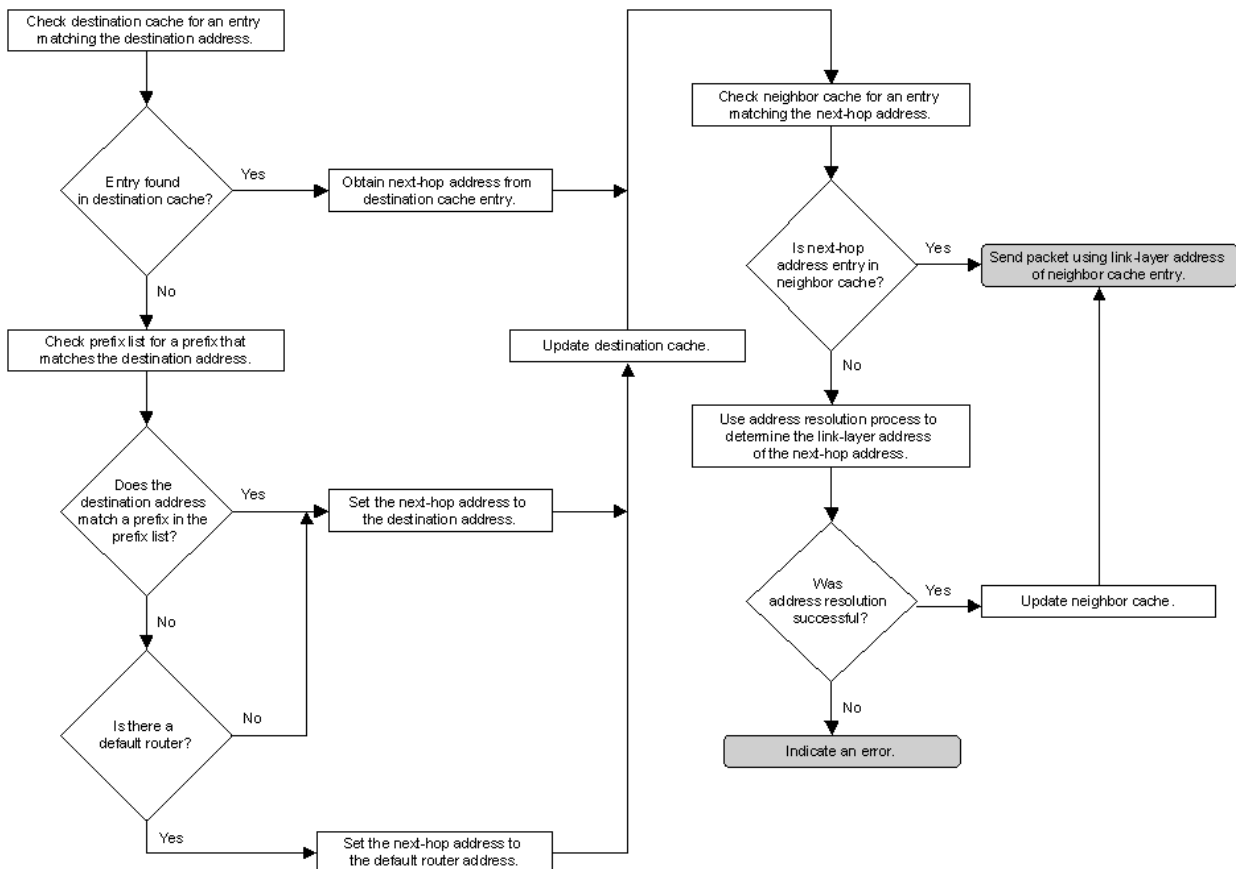


Figure 64 The host sending algorithm

---

## Address Autoconfiguration

One of the most useful aspects of IPv6 is its ability to automatically configure itself, even without the use of a stateful configuration protocol such as Dynamic Host Configuration Protocol for IPv6 (DHCPv6). By default, an IPv6 host can configure a link-local address for each interface. By using router discovery, a host can also determine the addresses of routers, other configuration parameters, additional addresses, and on-link prefixes. Included in the Router Advertisement message is an indication of whether a stateful address configuration protocol should be used.

Address autoconfiguration can only be performed on multicast-capable interfaces. Address autoconfiguration is described in RFC 4862.

### Autoconfigured Address States

Autoconfigured addresses are in one or more of the following states:

- Tentative

The address is in the process of being verified as unique. Verification is done through duplicate address detection. A node cannot receive unicast traffic to a tentative address. It can, however, receive and process multicast Neighbor Advertisement messages sent in response to the Neighbor Solicitation message that has been sent during duplicate address detection.

- Valid

An address for which uniqueness has been verified and from which unicast traffic can be sent and received. The valid state covers both the preferred and deprecated states. The amount of time that an address remains in the tentative and valid states is determined by the Valid Lifetime field in the Prefix Information option of a Router Advertisement message. The valid lifetime must be greater than or equal to the preferred lifetime. A valid address is either preferred or deprecated.

- Preferred

A node can send and receive unicast traffic to and from a preferred address. The period of time that an address can remain in the tentative and preferred states is determined by the Preferred Lifetime field in the Prefix Information option of a Router Advertisement message.

- Deprecated

An address that is still valid, but its use is discouraged for new communication. Existing communication sessions can still use a deprecated address. A node can send and receive unicast traffic to and from a deprecated address.

- Invalid

An address for which a node can no longer send or receive unicast traffic. An address enters the invalid state after the valid lifetime expires.

The relationship between the states of an autoconfigured address and the preferred and valid lifetimes is shown in Figure 65.

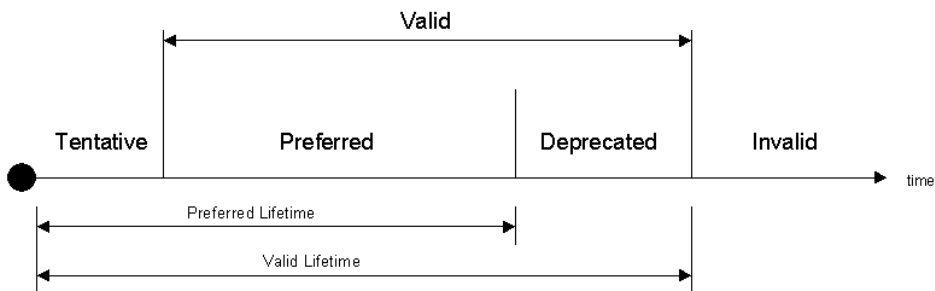


Figure 65 The states and lifetimes for an autoconfigured address

#### Note

With the exception of autoconfiguration for link-local addresses, address autoconfiguration is only specified for hosts. Routers must obtain address and configuration parameters through another means, such as manual configuration.

## Types of Autoconfiguration

There are three types of autoconfiguration:

### 1. Stateless

Configuration of addresses is based on the receipt of Router Advertisement messages with the Managed Address Configuration and Other Stateful Configuration flags set to 0 and one or more Prefix Information options.

### 2. Stateful

Configuration is based on the use of a stateful address configuration protocol such as DHCPv6 to obtain addresses and other configuration options. A host uses stateful address configuration when it receives Router Advertisement messages with no prefix options where either the Managed Address Configuration flag or the Other Stateful Configuration flag is set to 1. A host will also use a stateful address configuration protocol when there are no routers present on the local link.

### 3. Both

Configuration is based on receipt of Router Advertisement messages with Prefix Information options and the Managed Address Configuration or Other Stateful Configuration flags set to 1.

For all types, a link-local address is always configured.

The IPv6 protocol for Windows XP and Windows Server 2003 does not support the use of a stateful address configuration protocol such as DHCPv6. The IPv6 protocol for Windows Vista and Windows Server 2008 supports DHCPv6.

## Autoconfiguration Process

The address autoconfiguration process for the physical interface of an IPv6 node is the following:

1. A tentative link-local address is derived based on the link-local prefix of FE80::/64 and the 64-bit interface identifier.
2. Using duplicate address detection to verify the uniqueness of the tentative link-local address, a Neighbor Solicitation message is sent with the Target Address field that is set to the tentative link-

local address.

3. If a Neighbor Advertisement message sent in response to the Neighbor Solicitation message is received, this indicates that another node on the local link is using the tentative link-local address and address autoconfiguration stops. At this point, manual configuration must be performed on the node.
4. If no Neighbor Advertisement message (sent in response to the Neighbor Solicitation message) is received, the tentative link-local address is assumed to be unique and valid. The link-local address is initialized for the interface. The corresponding solicited-node multicast link-layer address is registered with the network adapter.

For an IPv6 host, the address autoconfiguration continues as follows:

1. The host sends up to 3 Router Solicitation messages (by default).
2. If no Router Advertisement messages are received, then the host uses a stateful address configuration protocol to obtain addresses and other configuration parameters.
3. If a Router Advertisement message is received, the Hop Limit, Reachable Time, Retrans Timer, and the MTU (if the MTU option is present) are set.

4. For each Prefix Information option present:

If the On-Link flag is set to 1, the prefix is added to the prefix list.

If the Autonomous flag is set to 1, the prefix and the 64-bit interface identifier are used to derive a tentative address.

Duplicate address detection is used to verify the tentative address's uniqueness.

If the tentative address is in use, the use of the address is not initialized for the interface.

If the tentative address is not in use, the address is initialized. This includes setting the valid and preferred lifetimes based on the Valid Lifetime and Preferred Lifetime fields in the Prefix Information option. It also includes registering the corresponding solicited-node multicast link-layer address with the network adapter.

5. If the Managed Address Configuration flag in the Router Advertisement message is set to 1, a stateful address configuration protocol is used to obtain additional addresses.
6. If the Other Stateful Configuration flag in the Router Advertisement message is set to 1, a stateful address configuration protocol is used to obtain additional configuration parameters.

The address autoconfiguration process for a host is shown in Figures 66 and 67.

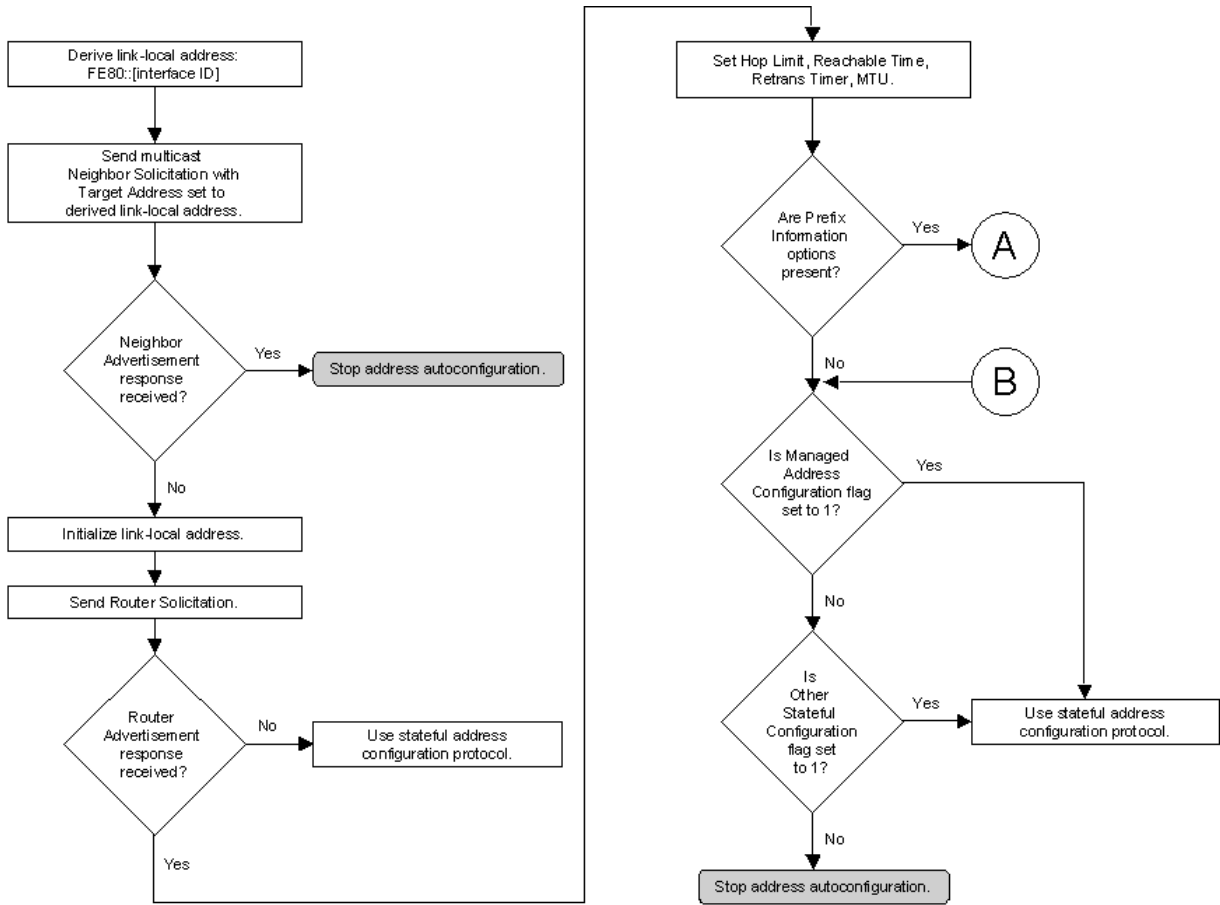


Figure 66 The address autoconfiguration process for a host (part 1)

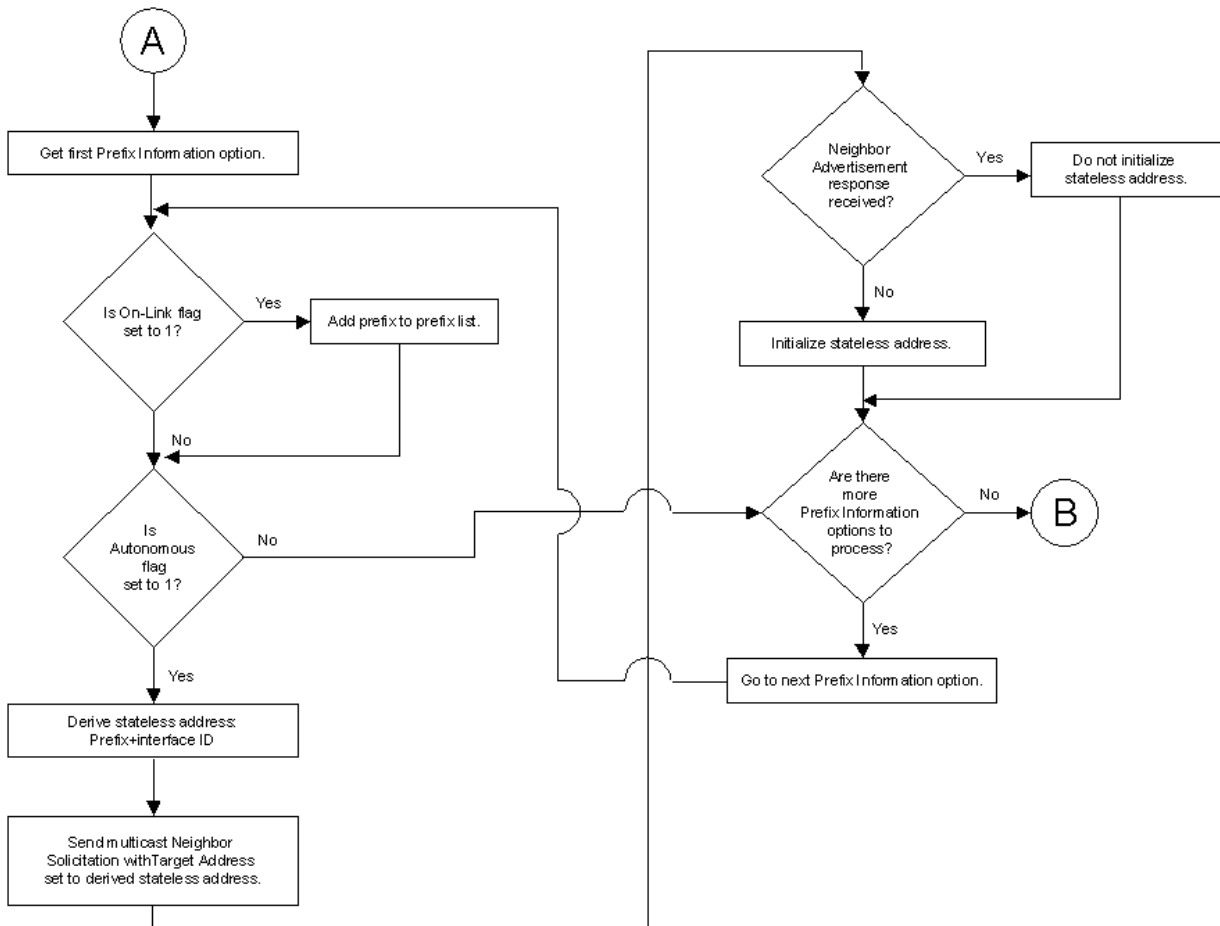


Figure 67 The address autoconfiguration process for a host (part 2)

## DHCPv6

DHCP for IPv6 (DHCPv6) is a protocol defined in RFC 3315 to perform stateful address configuration with IPv6 hosts or to provide stateless configuration information to IPv6 hosts. An IPv6 host performs stateless address autoconfiguration by default and stateful address autoconfiguration as indicated by the following fields in the Router Advertisement message sent by a neighboring router:

- **Managed Address Configuration flag** Also known as the M flag. When set, it instructs the host to use a stateful address configuration protocol to obtain a stateful address. This mode of operation is known as DHCPv6 stateful.
- **Other Stateful Configuration flag** Also known as the O flag. When set, it instructs the host to use a stateful address configuration protocol to obtain configuration settings.

It is possible for the M flag to be cleared and the O flag to be set. In this case, a DHCPv6-capable host does not use DHCPv6 to obtain addresses, but to obtain configuration settings. This mode of operation is known as DHCPv6 stateless. If either flag is set, a DHCPv6-capable host attempts to discover a DHCPv6 server to obtain address or configuration settings with a DHCPv6 message exchange.

## DHCPv6 Messages

Instead of broadcasting, a DHCPv6 client attempting to discover the location of the DHCPv6 server on the network sends a DHCPv6 message to the All\_DHCP\_Relay\_Agents\_and\_Servers address at FF02::1:2. If there is a DHCPv6 server on the host's subnet, it receives the message and sends an appropriate reply. More typically, a DHCPv6 relay agent on the host's subnet receives the message and forwards it to a DHCPv6 server. DHCPv6 servers and relay agents listen for DHCPv6 messages on UDP port 547. DHCPv6 clients listen on UDP port 546.

A DHCPv6 message exchange to obtain IPv6 addresses and configuration information typically consists of the following messages:

1. A Solicit message sent by the DHCPv6 client to locate the DHCPv6 servers. This is similar to the DHCPDiscover message used by DHCP for IPv4.
2. An Advertise message sent by a DHCPv6 server to indicate that it can provide address and configuration services. This is similar to the DHCPOffer message used by DHCP for IPv4.
3. A Request message sent by the DHCPv6 client to request address and configuration settings from a specific DHCPv6 server. This is similar to the DHCPRequest message used by DHCP for IPv4.
4. A Reply message sent by the DHCPv6 server that contains address and configuration settings. This is similar to the DHCPACK message used by DHCP for IPv4.

A DHCPv6 message exchange to obtain only configuration information typically consists of the following messages:

1. An Information-request message sent by the DHCPv6 client to request configuration settings from the DHCPv6 servers. This is similar to the DHCPInform message used by DHCP for IPv4.
2. A Reply message sent by a DHCPv6 server that contains the requested configuration settings.

For an IPv6 network that has routers configured to assign stateless address prefixes to IPv6 hosts, the 2-message DHCPv6 exchange can be used to assign DNS servers, DNS domain names, and other configuration settings that are not set through stateless address autoconfiguration.

## DHCPv6 Support in Windows

Windows Vista and Windows Server 2008 include a DHCPv6 client. The DHCPv6 client is enabled by default but only attempts DHCPv6-based configuration when indicated by the M and O flags in the received Router Advertisement message. Therefore, to use DHCPv6, you must configure your IPv6 routers to set these two flags at their appropriate values and then configure DHCPv6 servers and relay agents to service each IPv6 subnet.

Windows Server 2008 supports stateful and stateless DHCPv6 configuration and a DHCPv6 relay agent. You can configure the DHCP Server service for DHCPv6 scopes and options to be distributed to DHCPv6 clients in the four- or two-message DHCPv6 message exchange previously described.

---

## IPv6 Routing

Similar to IPv4 nodes, typical IPv6 nodes use a local IPv6 routing table to determine how to forward packets. IPv6 routing table entries are created by default when IPv6 initializes and additional entries are added either by the receipt of Router Advertisement messages containing on-link prefixes and routes, or through manual configuration.

An IPv6 routing table, which is present on all nodes running the IPv6 protocol for Windows, stores information about IPv6 address prefixes and how they are reached either directly or indirectly. Before the IPv6 routing table is checked, the destination cache is checked for an entry that matches the destination address in the IPv6 packet that is being forwarded. If an entry for the destination address is not in the destination cache, the routing table is used to determine:

1. The interface to be used for forwarding (the next-hop interface)

The interface identifies the physical or logical interface that is used to forward the packet to either its destination or to the next router.

2. The next-hop address

For a direct delivery (the destination is on a local link), the next-hop address is the destination address in the packet. For an indirect delivery (the destination is not on a local link), the next-hop address is the address of a router.

After the next-hop interface and address are determined, the destination cache is updated. Subsequent packets that are forwarded to the destination use the destination cache entry instead of checking the routing table.

### Contents of an IPv6 Routing Table

The following are the fields of a typical IPv6 routing table entry:

- Destination Prefix

The destination prefix is an IPv6 address prefix that can have a prefix length from 0 through 128. In the Windows IPv6 routing table, this column is named **Network Destination** or **Prefix**.

- Next-Hop Address

The address to which the packet is forwarded. In the Windows IPv6 routing table, this column is named **Gateway** or **Gateway/Interface Name**.

- Interface

The network interface that is used to forward the packet. All the addresses defined by the Destination Prefix are reachable over the interface. In the Windows IPv6 routing table, this column is named **If** or **Idx** (an abbreviation of Index) and stores the interface index of the forwarding interface.

- Metric

A number that is used to specify the cost of the route, so that the best route (potentially among multiple routes to the same destination) can be selected.

IPv6 routing table entries can be used to store the following types of routes:

- Directly-attached network routes

These routes are address prefixes for subnets that are directly attached (on-link) and typically have a prefix length of 64.

- Remote network routes

These routes are address prefixes for subnets that are not directly attached but are available through other routers. Remote network routes can be subnet prefixes (typically with a prefix length of 64) or a prefix for an address space (typically with a prefix length less than 64).

- Host routes

A host route is a route to a specific IPv6 address. Host routes allow routing to occur on a per-IPv6 address basis. For host routes, the route prefix is a specific IPv6 address with a prefix length of 128. In contrast, both types of network routes have prefixes that have a prefix length less than 128.

- Default route

The default route is used when a more specific network or host route is not found. The default route prefix is `::/0`.

## Route Determination Process

To determine which routing table entry is used to send or forward a packet, IPv6 uses the following process:

1. For a sending host, if the source address is specified by the sending application, the only routes that are checked are those that apply to the interface assigned the source address. For a sending host, if the source address is not specified by the sending application, or for a forwarding router, all the routes are checked. For each routing table entry that is checked, IPv6 compares the bits in the network prefix to those in the destination address for the number of bits specified in the prefix length of the route. For the number of bits in the prefix length, if all the bits in the network prefix match all the bits in the destination IPv6 address, the route is a match for the destination.
2. The list of matching routes is compiled. The route that has the largest prefix length (that is, the route that matched the most high-order bits with the destination address) is chosen. The longest matching route is the most specific route to the destination. If multiple entries with the longest match are found (for example, multiple routes to the same network prefix), the router uses the lowest metric to select the best route. If multiple entries exist that are the longest match and the lowest metric, IPv6 chooses which routing table entry to use.

For the IPv6 protocol for Windows Vista and Windows Server 2008, when there are multiple closest matching routes with the lowest metric, an on-link route is preferred over a route that is available through a router. Additionally, IPv6 will prefer routes that are available over supposedly reachable gateways over routes that are available over supposedly unreachable gateways. IPv6 can break ties between two supposedly unreachable gateways by alternating each lookup (each time a destination cache entry is created). If there are still multiple routes, IPv6 selects the first route in the table. For more information about the router selection process, see RFC 4191.

For any given destination, the above procedure results in finding matching routes in the following order:

1. A host route that matches the entire destination address
2. A network route with the longest prefix length that matches the destination
3. The default route (the prefix `::/0`)

The result of the route determination process is the selection of a single route in the routing table. The selected route yields a next-hop interface and address. The next-hop interface is the interface that is specified in the matching route. For remote traffic, the next-hop address is the address stored in the Next-Hop Address field (the address of a neighboring router). For traffic to neighbors on a directly-attached link, the next-hop address is the destination address of the packet. (In this case, an address is not stored in the Next-Hop Address field.)

If the route determination process on the sending host fails to locate a matching route, IPv6 treats the destination as locally reachable. If the route determination process on a router fails to locate a matching route, IPv6 sends an ICMPv6 Destination Unreachable-No Route to Destination message to the sending host and discards the packet.

## Example IPv6 Routing Tables for Windows Vista and Windows Server 2008

On a computer running Windows, you can display the IPv6 routing table using the following:

- The **route print** command
- The **netsh interface ipv6 show route** command

### The Route Print Command

The following is the IPv6 portion of an example display of the **route print** command for Windows Server 2008:

#### IPv6 Route Table

```
=====
```

#### Active Routes:

If	Metric	Network	Destination	Gateway
10	286	::/0		fe80::20a:42ff:feb0:5400
1	306	::/128		On-link
10	286	2001:db8::/64		On-link
10	286	2001:db8::/128		On-link
10	286	fe80::/64		On-link
10	286	fe80::251f:c754:e525:f20e/128		On-link
1	306	ff00::/8		On-link
10	286	ff00::/8		On-link

```
=====
```

#### Persistent Routes:

None

The display lists the individual routes, which can be categorized as the following:

- Routes with a 128-bit prefix length (`/128`) are host routes for a specific IPv6 destination. By default, only host routes for locally configured IPv6 address are in the IPv6 route table.

- Routes with a 64-bit prefix length (/64) are subnet routes for locally attached subnets.
- The ::/0 routes are default routes.
- The ff00::/8 are routes for multicast traffic.

### The netsh interface ipv6 show route Command

The **netsh interface ipv6 show route** command displays the IPv6 route table and includes information about whether the routes are published (if the computer is acting as an advertising router) and the route type. The following is an example of the **netsh interface ipv6 show route** command on a computer running Windows Vista:

Publish	Type	Met	Prefix	Idx	Gateway/Interface Name
No	0	0	:: /0	6	fe80::20a:42ff:feb0:5400
No	Manual	1	:: 1/128	1	Loopback Pseudo-Interface 1
No	0	0	3ffe:ffff:21da:7:: /64	6	Local Area Connection
No	Manual	1	3ffe:ffff:21da:7:1f3e:9e51:2178:b90b/128	6	Local Area Connection
No	Manual	1	3ffe:ffff:21da:7:a299:85ae:21da:59cc/128	6	Local Area Connection
No	Manual	1	fe80:: /64	6	Local Area Connection
No	Manual	1	fe80:: /64	10	Local Area Connection* 7
No	Manual	1	fe80:: /64	9	Local Area Connection* 6
No	Manual	1	fe80::5efe:1.0.0.127/128	10	Local Area Connection* 7
No	Manual	1	fe80::5efe:1.0.0.127/128	9	Local Area Connection* 6
No	Manual	1	fe80::713e:a426:d167:37ab/128	6	Local Area Connection

Each entry in the IPv6 routing table has the following fields:

- **Publish** specifies whether the route is published (advertised in a Routing Advertisement message).
- **Type** specifies the type of route.
- **Met** specifies the metric used to select between multiple routes with the same prefix.
- **Prefix** specifies the address prefix.
- **Idx** specifies the interface index, which indicates the interface over which packets that match the address prefix are reachable.

Interface indexes can be viewed from the display of the **netsh interface ipv6 show interface** command.

- **Gateway/Interface Name** specifies either a next-hop IPv6 address or an interface name.

For remote network routes, a next-hop IPv6 address is listed. For directly-attached network routes, the name of the interface from which the address prefix is directly reachable is listed.

The IPv6 routing table is built automatically, based on your IPv6 configuration. You can also add routes using the **netsh interface ipv6 add route** command.

An IPv6 router not only forwards IPv6 packets between interfaces, but also advertises both its presence and stateless address autoconfiguration information for hosts located on directly-attached subnets by sending Router Advertisement messages. A computer running Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008 can be configured as a static router and as an advertising router. A static router does not use routing protocols to maintain the routes in the IPv6 route table. For information about how to configure a computer running Windows as a static IPv6 router, see [Manual Configuration for IPv6](http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx>.

---

## Summary

This paper discussed the new IPv6 protocol suite by comparing, where possible, the IPv6 protocol suite to similar features or concepts that currently exist in IPv4. This paper discussed how IPv6 resolves IPv4 protocol design issues, the new IPv6 header and extension headers, ICMPv6 (the replacement for ICMP for IPv4), MLD (the replacement for IGMP for IPv4), IPv6 Neighbor Discovery processes that manage interaction between neighboring IPv6 nodes, IPv6 address autoconfiguration, and IPv6 routing. While not in prevalent use today, the future of the Internet will be IPv6-based. It is important to gain an understanding of this strategic protocol to begin planning for the eventual transition to IPv6.

---

## Related Links

See the following resources for further information:

- ["Understanding IPv6, Second Edition" Microsoft Press book](http://www.microsoft.com/MSPress/books/11607.aspx) at <http://www.microsoft.com/MSPress/books/11607.aspx>
- [Microsoft IPv6 Web site](http://www.microsoft.com/ipv6) at <http://www.microsoft.com/ipv6>
- [IP Version 6 Working Group Web site](http://www.ietf.org/html.charters/old/ipv6-charter.html) at <http://www.ietf.org/html.charters/old/ipv6-charter.html>

For the latest information about Windows Server 2008, see the [Windows Server 2008 Web site](http://www.microsoft.com/windowsserver2008) at <http://www.microsoft.com/windowsserver2008>.



Windows Server System is comprehensive, integrated, and interoperable server infrastructure that simplifies the development, deployment, and management of flexible business solutions.  
[www.microsoft.com/windowsserversystem](http://www.microsoft.com/windowsserversystem)