



IPv6 for Cisco IOS Software, File 1 of 3: Overview

This document provides an overview of the Cisco implementation of IP version 6 (IPv6) in the Cisco IOS software and includes the following sections:

- [Documentation Specifics, page 2](#)
- [Feature Overview—IPv6, page 2](#)
 - [Larger Address Space, page 2](#)
 - [Simplified Packet Header, page 9](#)
 - [ICMP for IPv6, page 13](#)
 - [Neighbor Discovery, page 14](#)
 - [DNS for IPv6, page 19](#)
 - [Path MTU Discovery, page 19](#)
 - [Standard Access Control Lists, page 20](#)
 - [SSH over an IPv6 Transport, page 20](#)
 - [TFTP, ping, Telnet, and traceroute, page 20](#)
 - [Data Link Layer Protocols, page 21](#)
 - [Routing Protocols, page 21](#)
 - [Distributed CEF Switching for IPv6, page 22](#)
- [Benefits, page 23](#)
 - [Stateless Autoconfiguration, page 23](#)
 - [Simplified Network Renumbering for IPv6 Hosts, page 24](#)
 - [Prefix Aggregation, page 24](#)
 - [Site Multihoming, page 25](#)
 - [Mobile IP, page 25](#)
 - [Security, page 26](#)
 - [Transition Richness, page 26](#)
- [Glossary, page 29](#)

Documentation Specifics

IPv6 features are supported only in the 12.0 ST and 12.2 T Cisco IOS software release trains, starting at Cisco IOS Release 12.0(21)ST and Release 12.2(2)T, respectively. Subsequent releases of the 12.0 ST and 12.2 T Cisco IOS software trains will support additional IPv6 features. This document, along with the following IPv6 for Cisco IOS Software feature documentation in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com, will be updated with information about additional IPv6 features at each subsequent release of the applicable Cisco IOS software release trains:

- *Start Here: Cisco IOS Software Release Specifics for IPv6 Features*
- *IPv6 for Cisco IOS Software, File 1 of 3: Overview* (This document)
- *IPv6 for Cisco IOS Software, File 2 of 3: Configuring*
- *IPv6 for Cisco IOS Software, File 3 of 3: Commands*



Note

The *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document details which IPv6 features are supported in each release of the 12.0 ST and 12.2 T Cisco IOS software trains. *Not all IPv6 features may be supported in your Cisco IOS software release.* We strongly recommend that you read the entire *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document before reading the other IPv6 for Cisco IOS Software feature documentation.

The other IPv6 for Cisco IOS Software feature documentation provides IPv6 overview, configuration, and command reference information for IPv6 features in each release of the 12.0 ST and 12.2 T Cisco IOS software trains.

Feature Overview—IPv6

The Internet Protocol (IP) is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IP handles addressing, fragmentation, reassembly, and protocol demultiplexing. It is the foundation on which all other IP protocols (collectively referred to as the IP Protocol suite) are built. As a network-layer protocol, IP contains addressing and control information that allows data packets to be routed. IPv6, formerly called IPng (next generation), is the latest version of IP that offers many benefits, such as a larger address space, over the previous version of IP (version 4).

Larger Address Space

The primary motivation for IPv6 is the need to meet the anticipated future demand for globally unique IP addresses. Applications such as mobile Internet-enabled devices (such as personal digital assistants [PDAs], telephones, and cars), home-area networks (HANs), and wireless data services are driving the demand for globally unique IP addresses. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. By being globally unique, IPv6 addresses inherently enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for the addresses. Additionally, the flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT); therefore, IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x. Following are two examples of IPv6 addresses:

```
FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
```

```
1080:0:0:0:8:800:200C:417A
```

It is common for IPv6 addresses to contain successive hexadecimal fields of zeros. To make IPv6 addresses less cumbersome, two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). [Table 1](#) lists compressed IPv6 address formats.

A double colon may be used as part of the *ipv6-address* argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.



Note

Two colons (::) can only be used once in an IPv6 address.

The hexadecimal letters in IPv6 addresses are not case sensitive.

Table 1 *Compressed IPv6 Address Formats*

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	1080:0:0:0:8:800:200C:417A	1080::8:800:200C:417A
Multicast	FF01:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0	::

The loopback address listed in [Table 1](#) may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).



Note

The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

The unspecified address listed in [Table 1](#) indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.



Note

The IPv6 unspecified address cannot be assigned to an interface. Unspecified IPv6 addresses should not be used as destination addresses in IPv6 packets or the IPv6 routing header.

An IPv6 address prefix, in the format *ipv6-prefix/prefix-length*, can be used to represent bit-wise contiguous blocks of the entire address space. The *ipv6-prefix* variable must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The */prefix-length* variable is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 1080:6809:8086:6502::/64 is an acceptable IPv6 prefix.

IPv6 Address Types

Following are the three types of IPv6 addresses:

- Unicast—An address for a single interface. A packet that is sent to a unicast address is delivered to the interface identified by that address. The Cisco IOS software supports the following IPv6 unicast address types:
 - Global aggregatable address
 - Site-local address
 - Link-local address
 - IPv4-compatible IPv6 address
- Anycast—An address for a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address.
- Multicast—An address for a set of interfaces (in a given scope) that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address (in a given scope).



Note

There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.



Note

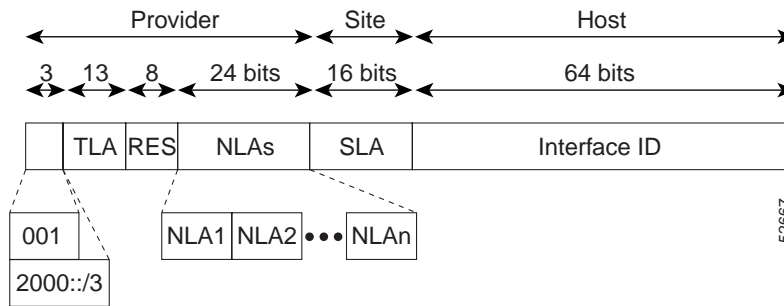
In Cisco IOS Release 12.2(4)T or later releases and Cisco IOS Release 12.0(21)ST or later releases, multiple IPv6 global and site-local addresses within the same prefix can be configured on an interface. Multiple IPv6 link-local addresses on an interface are not supported.

Refer to the *IPv6 for Cisco IOS Software, File 2 of 3: Configuring* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for information on configuring IPv6 addressing and enabling IPv6 routing. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Aggregatable Global Address

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses used on links are aggregated upward through organizations, then to intermediate-level Internet service providers (ISPs), and eventually to top-level ISPs. [Figure 1](#) shows the structure of an aggregatable global address.

Figure 1 *Aggregatable Global Address Format*



A fixed prefix of 2000::/3 (001) indicates an aggregatable global IPv6 address. Addresses with a prefix of 2000::/3 (001) through E000::/3 (111), excluding the FF00::/8 (1111 1111) multicast addresses, are required to have 64-bit interface identifiers in the modified EUI-64 format.

A Top-Level Aggregator (TLA) identifies tier 1 ISPs. TLAs are connected in a default-free zone. Routers in the default-free zone must have a default-free routing table entry for every active TLA identifier.

A field of 8 bits is a reserved field for the growth of the TLA and Next-Level Aggregator (NLA) fields. The reserved field must always be equal to zero.

An NLA identifies intermediate service providers assigned an NLA to create an addressing hierarchy and to identify sites. An organization can assign the top part of the NLA in a manner to create an addressing hierarchy appropriate to its network. It can use the remainder of the bits in the field to identify sites it wants to serve.

A Site-Level Aggregator (SLA) is used by individual organizations to create their own local addressing hierarchy and to identify subnets. An SLA is similar to a subnet in IPv4, except that an organization with an SLA has a much greater number of subnets to utilize; the 16-bit SLA field supports 65,535 individual subnets.

An interface identifier is used to identify interfaces on a link. The interface identifier must be unique to the link. They may also be unique over a broader scope. In many cases, an interface identifier will be the same as or based on the link-layer address of an interface. Interface identifiers used in aggregatable global unicast and other IPv6 address types must be 64 bits long and constructed in the modified EUI-64 format.

Interface identifiers are constructed in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and FDDI interfaces), the first three octets (24 bits) are taken from the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are taken from the last three octets of the MAC address. The construction of the interface identifier is completed by setting the Universal/Local (U/L) bit—the seventh bit of the first octet—to a value of 0 or 1. A value of 0 indicates a locally administered identifier; a value of 1 indicates a globally unique IPv6 interface identifier.
- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel—except tunnel interfaces used with IPv6 overlay tunnels—interface types), the interface identifier is constructed in the same way as the interface identifier for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used to construct the identifier (the MAC address from the interface is not used).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface identifier is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.



Note For interfaces using PPP, given that the interfaces at both ends of the connection might have the same MAC address, the interface identifiers used at both ends of the connection are negotiated (picked randomly and, if necessary, reconstructed) until both identifiers are unique. Given that the first MAC address in the router is used to construct the identifier for interfaces using PPP, it is unlikely that the interface identifiers used at both ends of the connection would not be unique.

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

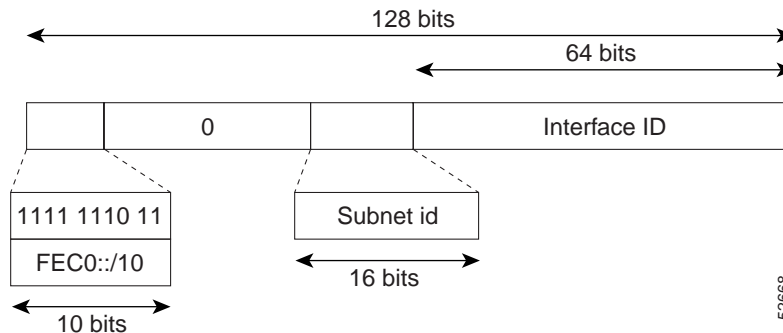
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash (a “message digest”) to determine the MAC address of the router from the host name of the router.

Site-Local Address

A site-local address is an IPv6 unicast address that uses the prefix FEC0::/10 (1111 1110 11) and concatenates the subnet identifier (the 16-bit SLA field) with the interface identifier in the modified EUI-64 format. Site-local addresses can be used to number a complete site without using a globally unique prefix. Site-local addresses can be considered private addresses because they can be used to restrict communication to a limited domain. [Figure 2](#) shows the structure of a site-local address.

IPv6 routers must not forward packets that have site-local source or destination addresses outside of the site.

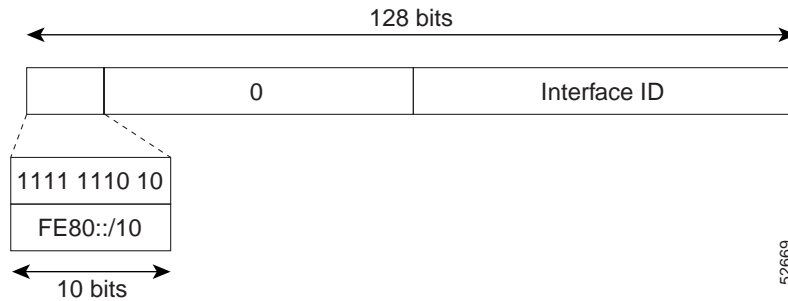
Figure 2 Site-Local Address Format



Link-Local Address

A link-local address is an IPv6 unicast address that can be automatically configured on any interface by using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need site-local or globally unique addresses to communicate. [Figure 3](#) shows the structure of a link-local address.

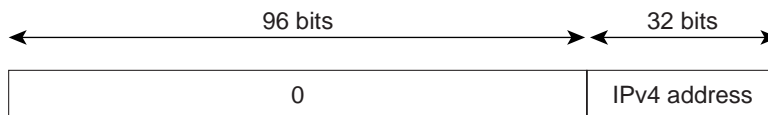
IPv6 routers must not forward packets that have link-local source or destination addresses to other links.

Figure 3 Link-Local Address Format

52669

IPv4-Compatible IPv6 Address

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is `0:0:0:0:0:A.B.C.D` or `::A.B.C.D`. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32-bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels. Figure 4 shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 4 IPv4-Compatible IPv6 Address Format

`0:0:0:0:0:192.168.30.1`
`::192.168.30.1`
`::C0A8:1E01`

52727

Anycast Address

An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface makes a unicast address an anycast address. Nodes to which the anycast address is assigned must be explicitly configured to recognize that the address is an anycast address.



Note

Anycast addresses must not be used as the source address of an IPv6 packet.

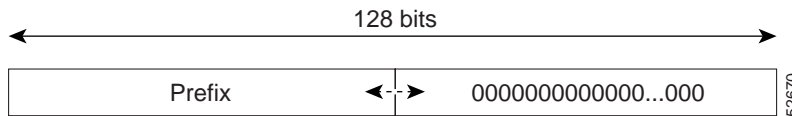


Note

Cisco networking devices can process packets that are destined for anycast addresses; however, in the current Cisco IOS software releases that support IPv6, anycast addresses cannot be configured on network interfaces in Cisco networking devices. The configuring of anycast addresses on network interfaces in Cisco networking devices will be supported in a later release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Figure 5 shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

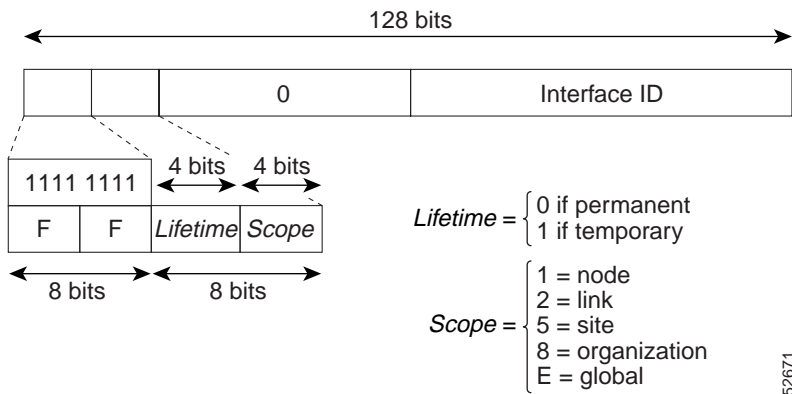
Figure 5 Subnet Router Anycast Address Format



IPv6 Multicast Address

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. Figure 6 shows the format of the IPv6 multicast address.

Figure 6 IPv6 Multicast Address Format



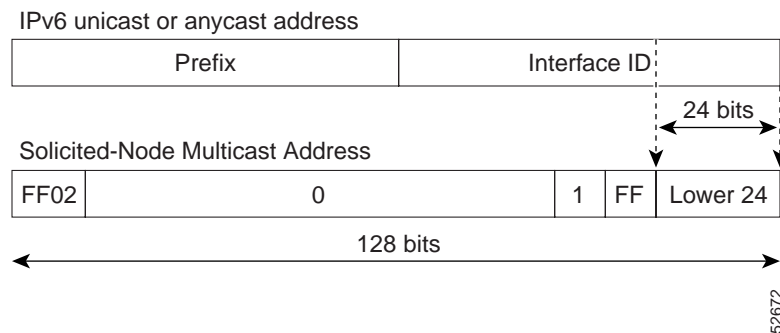
IPv6 nodes (hosts and routers) are required to join (receive packets destined for) the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:1 (scope is link-local)
- Solicited-node multicast group FF02:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:2 (scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address. (See [Figure 7](#).) For example, the solicited-node multicast address corresponding to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

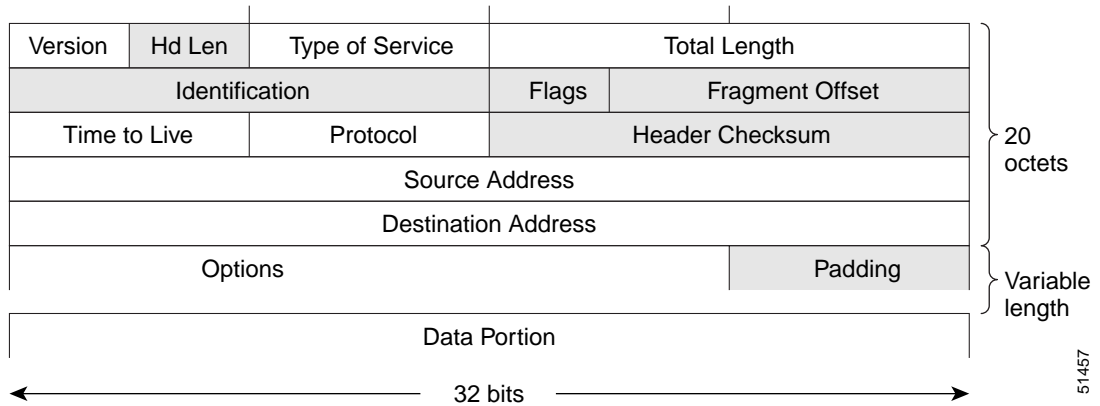
Figure 7 IPv6 Solicited-Node Multicast Address Format



Simplified Packet Header

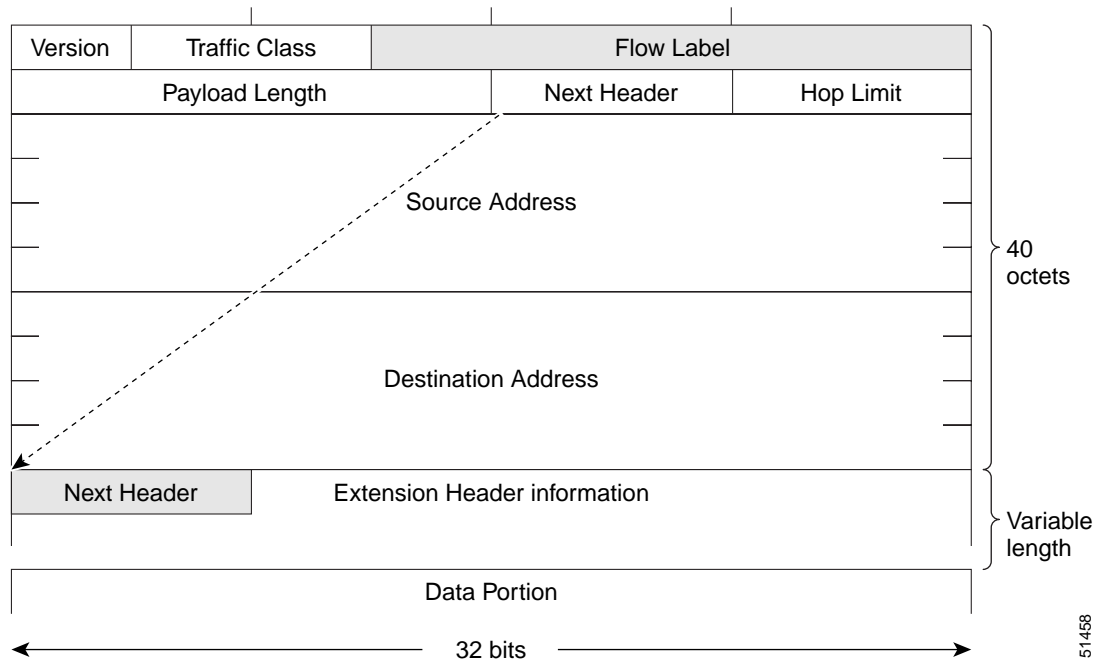
The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits). (See [Figure 8](#).) The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header shown in [Figure 8](#) are not included in the IPv6 packet header.

Figure 8 IPv4 Packet Header Format



The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). (See [Figure 9](#).) Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by routers and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. (In IPv4, the User Datagram Protocol (UDP) transport layer uses an optional checksum. In IPv6, use of the UDP checksum is required to check the integrity of the inner packet.) Additionally, the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

Figure 9 IPv6 Packet Header Format



[Table 2](#) lists the fields in the basic IPv6 packet header.

Table 2 Basic IPv6 Packet Header Fields

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information following the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in Figure 9 .
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Following the eight fields of the basic IPv6 packet header are optional extension headers and the data portion of the packet. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Together, the extension headers form a chain of headers. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. [Figure 10](#) shows the IPv6 extension header format.

Figure 10 IPv6 Extension Header Format

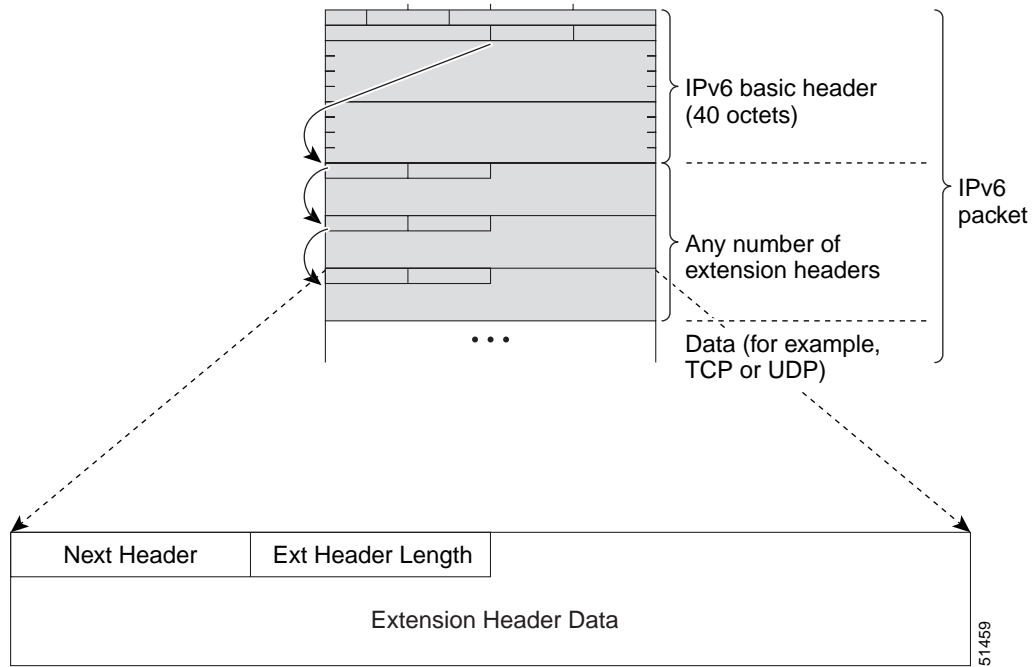


Table 3 lists the extension header types and their Next Header field values.

Table 3 IPv6 Extension Header Types

Header Type	Next Header Value	Description
Hop-by-hop options header	0	This header is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.
Destination options header	60	The destination options header can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination.
Routing header	43	The routing header is used for source routing.
Fragment header	44	The fragment header is used when a source must fragment a packet that is larger than the Maximum Transmission Unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.

Table 3 IPv6 Extension Header Types (continued)

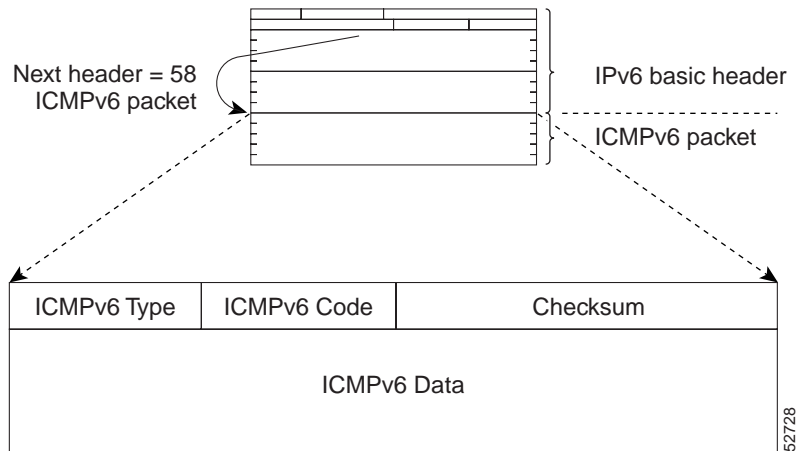
Header Type	Next Header Value	Description
Authentication header and ESP header	51	The Authentication header and the ESP header are used within IP Security Protocol (IPSec) to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.
Upper-layer header	6 (TCP) 17 (UDP)	The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP and UDP.

ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4—ICMP generates error messages, such as ICMP destination unreachable messages and informational messages like ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet is after all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. [Figure 11](#) shows the IPv6 ICMP packet header format.

Figure 11 IPv6 ICMP Packet Header Format



Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighbor routers.

The Static Cache Entry for IPv6 Neighbor Discovery feature enables the configuring of static entries in the IPv6 neighbor discovery cache, which provides functionality in IPv6 that is equivalent to static Address Resolution Protocol (ARP) entries in IPv4. Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process. The Cisco IOS software uses static ARP entries in IPv4 to translate 32-bit IP addresses into 48-bit hardware addresses. In IPv6, the Cisco IOS software uses static entries in the IPv6 neighbor discovery cache to translate 128-bit IPv6 addresses into 48-bit hardware addresses. Refer to the “Configuring Static Cache Entries for IPv6 Neighbor Discovery” section in the *IPv6 for Cisco IOS Software, File 2 of 3: Configuring* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for information on configuring static cache entries for IPv6 neighbor discovery.

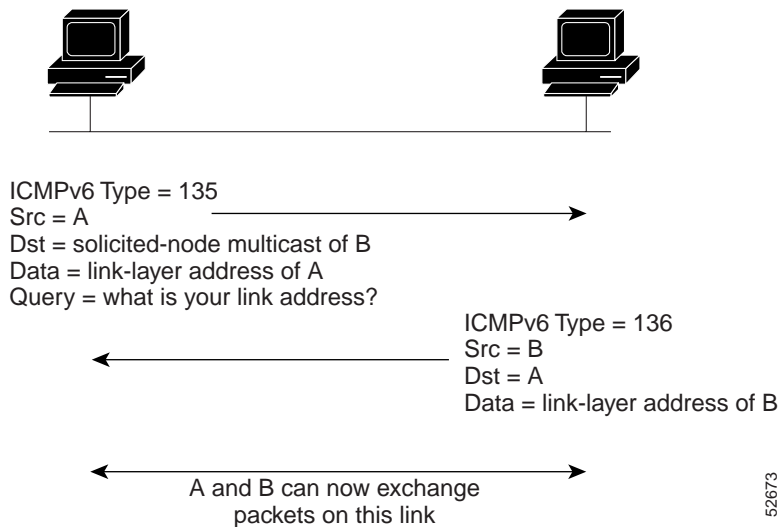


Note

Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link. (See [Figure 12](#).) When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 12 IPv6 Neighbor Discovery—Neighbor Solicitation Message



After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node's interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor solicitation message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verify the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor, and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination) or the receipt of a neighbor advertisement message in response to a neighbor solicitation message. If packets are reaching the peer, they are also reaching the next hop neighbor of the source. Therefore, forward progress is also a confirmation that the next hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

**Note**

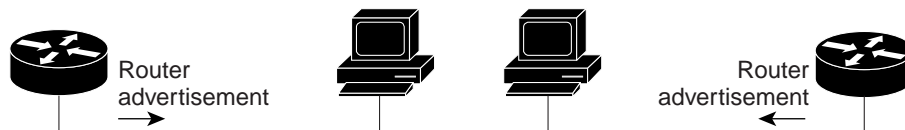
A neighbor advertisement message that has the solicited flag set to a value of 0 must not be considered as a positive acknowledgment that the forward path is still working.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). Specifically, a node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

Every IPv6 unicast address (global or link-local) must be checked for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address. The Cisco implementation of duplicate address detection in the Cisco IOS software does not check the uniqueness of anycast or global addresses that are generated from 64-bit interface identifiers.

Router advertisement messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 router. The router advertisement messages are sent to the all-nodes multicast address. (See [Figure 13](#).)

Figure 13 IPv6 Neighbor Discovery—Router Advertisement Message



Router advertisement packet definitions:
 ICMPv6 Type = 134
 Src = router link-local address
 Dst = all-nodes multicast address
 Data = options, prefix, lifetime, autoconfig flag

52674

Router advertisement messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Lifetime information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or statefull) that can be completed
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates

Router advertisements are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. Given that router solicitation messages are usually sent by hosts at system startup (the host does not have a configured unicast address), the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When a router advertisement is sent in response to a router solicitation, the destination address in the router advertisement message is the unicast address of the source of the router solicitation message.

The following router advertisement message parameters can be configured:

- The time interval between periodic router advertisement messages
- The “router lifetime” value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of router advertisement messages (with default values) is automatically enabled on Ethernet and FDDI interfaces when the **ipv6 unicast-routing** global configuration command is configured. For other interface types, the sending of router advertisement messages must be manually configured by using the **no ipv6 nd suppress-ra** global configuration command. The sending of router advertisement messages can be disabled on individual interfaces by using the **ipv6 nd suppress-ra** global configuration command.

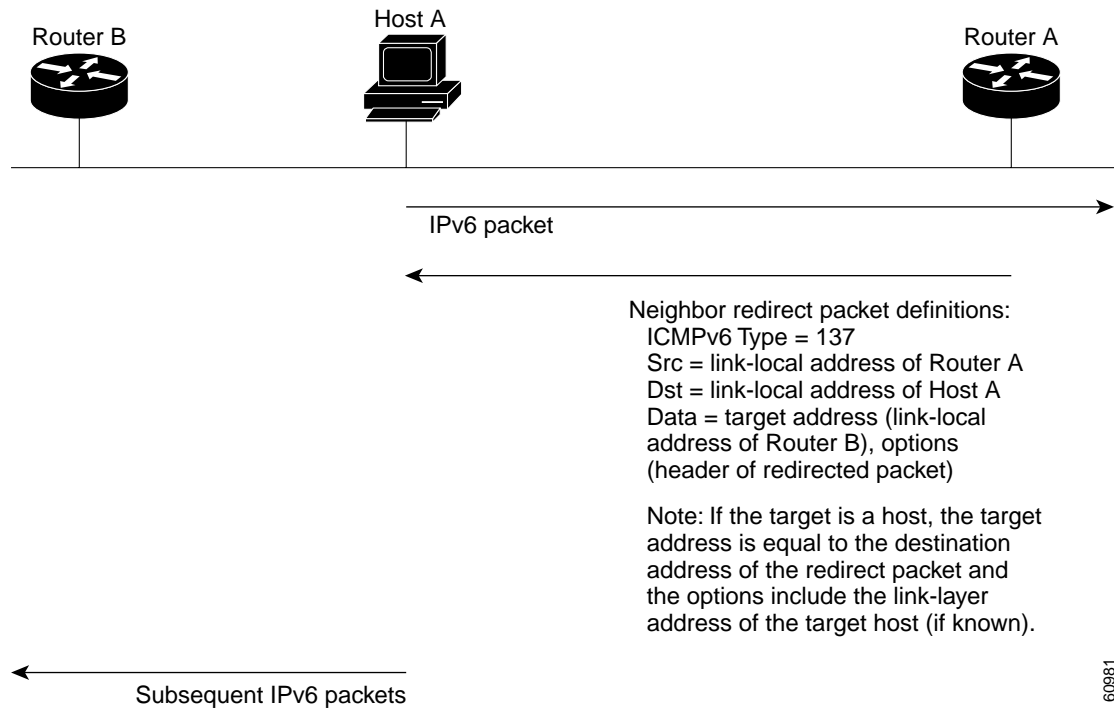
**Note**

For stateless autoconfiguration to work properly, the advertised prefix length in router advertisement messages must always be 64 bits.

For more information on the **ipv6 unicast-routing** and **ipv6 nd suppress-ra** commands, refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Routers send neighbor redirect messages to inform hosts of better first hop nodes on the path to a destination. (See [Figure 14](#).)

Figure 14 IPv6 Neighbor Discovery—Neighbor Redirect Message



Note

A router must be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor router by its link-local address. For static routing, the address of the next hop router should be specified using the link-local address of the router; for dynamic routing, all IPv6 routing protocols must exchange the link-local addresses of neighboring routers.

After forwarding a packet, a router should send a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the router.
- The packet is about to be sent out the interface on which it was received.
- The router determines that a better first hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link, or a link-local address.

Use the **ipv6 icmp error-interval** command to limit the rate at which the router generates all IPv6 ICMP error messages, including neighbor redirect messages, which ultimately reduces link-layer congestion.



Note

A router must not update its routing tables after receiving a neighbor redirect message, and hosts must not originate neighbor redirect messages.

DNS for IPv6

IPv6 introduces new Domain Name System (DNS) record types that are supported in the DNS name-to-address and address-to-name lookup processes. The new DNS record types support IPv6 addresses. [Table 4](#) lists the new IPv6 DNS record types.

Table 4 IPv6 DNS Record Types

Record Type	Description	Format
AAAA	<p>Maps a host name to an IPv6 address. (Equivalent to an A record in IPv4.)</p> <p>Note Support for AAAA records and A records over an IPv6 transport or IPv4 transport is in the latest release of the Cisco IOS software. Refer to the <i>Start Here: Cisco IOS Software Release Specifics for IPv6 Features</i> document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.</p>	www.abc.test AAAA 3FFE:B00:C18:1::2
PTR	<p>Maps an IPv6 address to a host name. (Equivalent to a PTR record in IPv4.)</p> <p>Note The Cisco IOS software supports PTR records for the IP6.INT domain.</p>	2.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.0.0.b.0.e.f.f.3.ip6.int PTR www.abc.test

Path MTU Discovery

As in IPv4, path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of every link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently.



Note

In IPv4, the minimum link MTU is 68 octets, which means the MTU size of every link along a given data path must support an MTU size of at least 68 octets.

In IPv6, the minimum link MTU is 1280 octets. We recommend using an MTU value of 1500 octets for IPv6 links.

Standard Access Control Lists

IPv6 standard Access Control Lists (ACLs) are used for basic traffic filtering functions. As in IPv4, IPv6 ACLs filter traffic based on source and destination addresses, inbound and outbound to a specific interface, and with an implicit deny at the end of an access list. Refer to the *IPv6 for Cisco IOS Software, File 2 of 3: Configuring* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for information on configuring IPv6 ACLs. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

SSH over an IPv6 Transport

In Cisco IOS Release 12.2(8)T or later releases, Secure Shell (SSH) in IPv6 functions the same and offers the same benefits as SSH in IPv4—the SSH Server feature enables an SSH client to make a secure, encrypted connection to a Cisco router and the SSH Client feature enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running an SSH server. IPv6 enhancements to SSH consist of support for IPv6 addresses that enable a Cisco router to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

There are no configuration tasks specifically for the SSH over an IPv6 Transport feature. After the SSH server is enabled by using the **ip ssh** global configuration command, the **ssh EXEC** command can be used to start an encrypted session with a remote IPv4 or IPv6 networking device.



Note

The SSH client runs in user EXEC mode and has no specific configuration tasks or commands. The SSH client is available only when the SSH server is enabled by using the **ip ssh** command.

Refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for command reference information for the SSH over an IPv6 Transport feature. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for IPv6 features.

Refer to the “Configuring Secure Shell” chapter in the Release 12.2, *Cisco IOS Security Configuration Guide* for additional SSH configuration information. Refer to the “Secure Shell Commands” chapter in the *Cisco IOS Security Command Reference* for additional SSH command information.

TFTP, ping, Telnet, and traceroute

IPv6 supports TFTP file downloading and uploading using the **copy EXEC** command. For example, to save the running configuration of the router to an IPv6 TFTP server, use the following command:

```
Router# copy running-config tftp://[3ffe:b00:c18:1:290:27ff:fe3a:9e9a]/running-config
```



Note

In Cisco IOS Release 12.2(8)T or later releases, a literal IPv6 address specified with a port number must be enclosed in square brackets ([]) when the address is used in Trivial File Transfer Protocol (TFTP) source or destination URLs; a literal IPv6 address specified without a port number need not be enclosed in square brackets. Refer to RFC 2732, *Format for Literal IPv6 Addresses in URL's*, for more information on the use of square brackets with literal IPv6 address in URLs.

The **ping** EXEC command accepts a destination IPv6 address or IPv6 host name as an argument and sends ICMPv6 echo request messages to the specified destination. The ICMPv6 echo reply messages are reported on the console. Extended ping functionality is also supported in IPv6.

The Telnet client and server in the Cisco IOS software support IPv6 connections. A user can directly Telnet to the router using an IPv6 Telnet client. An IPv6 Telnet connection can also be initiated from the router.

The **traceroute** EXEC command accepts a destination IPv6 address or IPv6 host name as an argument and will generate IPv6 traffic to report each IPv6 hop used to reach the destination address.


Note

In Cisco IOS Release 12.2(8)T or later releases, literal IPv6 addresses specified with other commands, such as **ping** and **traceroute**, may also be enclosed in square brackets.

Refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for information on IPv6 modifications to the **copy**, **ping**, **telnet**, and **traceroute** commands. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Data Link Layer Protocols

IPv6 protocol identifiers have been defined for most data link layer protocols. For example, IPv6 uses protocol ID 0x86DD for Ethernet packets. (See [Figure 15](#).) The protocol ID differentiates IPv6 Ethernet packets from Ethernet packets for other protocols.


Note

Packet encapsulation in Ethernet frames over IEEE 802.3 Ethernet LAN is not specified for IPv6.

Figure 15 IPv6 Protocol ID for Ethernet Packets

Destination Ethernet Address	Source Ethernet Address	0x86DD	IPv6 header and payload
------------------------------------	-------------------------------	--------	-------------------------

52675

In Cisco IOS Release 12.2(8)T or later releases, the CDP IPv6 Address Support for Neighbor Information feature adds the ability to transfer IPv6 addressing information between two Cisco devices using Cisco Discovery Protocol (CDP). CDP support for IPv6 addresses allows CDP to exchange IPv6 addressing information. CDP support for IPv6 addresses provides IPv6 information to network management products, and troubleshooting tools.

Routing Protocols

IPv6 supports Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). Routing Information Protocol (RIP) and Integrated Intermediate System-to-Intermediate System (IS-IS) are the supported IGPs for IPv6. Multiprotocol Border Gateway Protocol (BGP) is the supported EGP for IPv6.

When configuring supported routing protocols in IPv6, you must create the routing process, enable the routing process on interfaces, and customize the routing protocol for your particular network.

RIP in IPv6 functions the same and offers the same benefits as RIP in IPv4. IPv6 enhancements to RIP include support for IPv6 addresses and prefixes, and the use of the all RIP routers multicast group address FF02::9 as the destination address for RIP update messages. New commands specific to RIP in IPv6 were also added to the Cisco IOS command-line interface (CLI).

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and Open System Interconnection (OSI) routes. Extensions to the IS-IS CLI allow configuration of IPv6-specific parameters. IS-IS in IPv6 extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.

Multiprotocol BGP in IPv6 functions the same and offers the same benefits as multiprotocol BGP in IPv4. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses and scoped addresses (the next hop attribute uses a global IPv6 address and potentially also a link-local address, when a peer is reachable on the local link).

Refer to the *IPv6 for Cisco IOS Software, File 2 of 3: Configuring* and *IPv6 for Cisco IOS Software, File 3 of 3: Commands* documents in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for configuration and command reference information on IPv6 RIP and IPv6 multiprotocol BGP. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for a list of IPv6 RIP, IPv6 IS-IS and IPv6 multiprotocol BGP RFCs, and Cisco IOS software release specifics for IPv6 features.

Distributed CEF Switching for IPv6

Cisco Express Forwarding for IPv4 (CEFv4) is advanced, Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet, on networks characterized by intensive web-based applications or interactive sessions. CEFv4 avoids the potential overhead of continuous cache churn by using a forwarding information base (FIB) for the destination switching decision mirroring the entire contents of the IP routing table. In CEFv4, the Route Processor (RP) performs the centralized switching operation.

Distributed CEF for IPv4 (dCEFv4) performs the same functions as CEF but for distributed architecture platforms such as the Cisco 12000 series Internet routers. Instead of the RP performing the switch operation, the line cards hold an identical copy of the FIB database, which allows them to autonomously perform express forwarding, thus relieving the RP.

In Cisco IOS Release 12.0(21)ST or later releases, the distributed CEF for IPv6 (dCEFv6) feature offers the same benefits as CEF and dCEF for IPv4, with the addition of meeting future demand for globally unique IP addresses that quadruple the number of network address bits from 32 bits (in IPv4) to 128 bits.

Refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* documents in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for command reference information on dCEFv6. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for a list of RFCs, and Cisco IOS software release specifics for IPv6 features.

Benefits

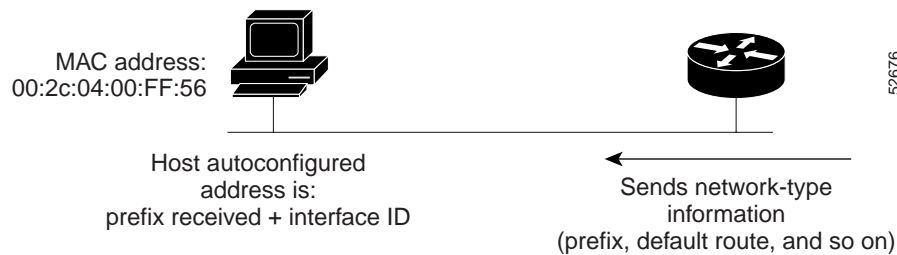
The following sections explain the benefits of IPv6.

Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::0. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

Nodes can connect to a network and automatically generate site-local and global IPv6 address without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) Server. With IPv6, a router on the link advertises in router advertisement messages any site-local and global prefixes, and its willingness to function as a default router for the link. Router advertisement messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup. (See [Figure 16](#).)

Figure 16 IPv6 Stateless Autoconfiguration



A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the router advertisement messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the router advertisement messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message.

Simplified Network Renumbering for IPv6 Hosts

The strict aggregation of the global routing table requires that networks be renumbered when the service provider for the network is changed. When the stateless autoconfiguration functionality in IPv6 is used to renumber a network, the prefix from a new service provider is added to router advertisement messages that are sent on the link. (The router advertisement messages contain both the prefix from the old service provider and the prefix from the new service provider.) Nodes on the link automatically configure additional addresses by using the prefix from the new service provider. The nodes can then use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. By configuring the lifetime parameters associated with the old and new prefixes, nodes on the link can make the transition to using only addresses created from the new prefix. During a transition period, the old prefix is removed from router advertisement messages and only addresses that contain the new prefix are used on the link (the renumbering is complete). (See [Figure 17](#).)

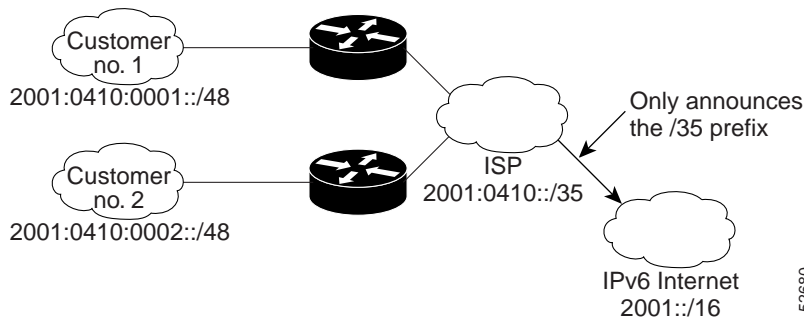
Figure 17 IPv6 Network Renumbering for Hosts Using Stateless Autoconfiguration



Prefix Aggregation

The aggregatable nature of the IPv6 address space enables an IPv6 addressing hierarchy. For example, an enterprise can subdivide a single IPv6 prefix from a service provider into multiple, longer prefixes for use within its internal network. Conversely, a service provider can aggregate all of the prefixes of its customers into a single, shorter prefix that the service provider can then advertise over the IPv6 internet. (See [Figure 18](#).)

Figure 18 IPv6 Prefix Aggregation



Site Multihoming

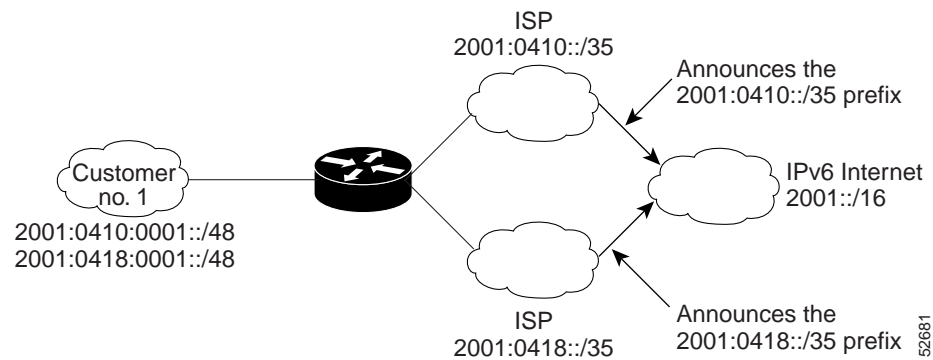
Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network makes it easy for that network to connect to multiple ISPs without breaking the global routing table. (See [Figure 19](#).)



Note

The source address selection process in both IPv6 and IPv4 multihomed networks is an ongoing research topic being discussed by the Internet Engineering Task Force (IETF).

Figure 19 IPv6 Site Multihoming



Mobile IP

Mobile IP provides users the freedom to roam beyond their home subnet while consistently maintaining their home IP address. Mobile IP enables transparent routing of IP datagrams to mobile users during their movement, so that data sessions can be initiated to them while they roam; it also enables sessions to be maintained in spite of physical movement between points of attachment to the Internet or other networks. The Cisco implementation of Mobile IP for IPv4 is fully compliant with the IETF proposed standard defined in RFC 2002, *IP Mobility Support*. The Cisco implementation of Mobile IP for IPv6 will be fully compliant with the IETF draft standard *Mobility Support in IPv6*.



Note

Support for Mobile IP in IPv6 is not in the current release of the Cisco IOS software. Mobile IP in IPv6 will be supported in a later release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

In IPv6, Mobile IP is implemented using the routing extension header. The routing extension header enables a mobile node to send IP packets directly to a destination node after the mobile node establishes an initial connection to the home agent. Direct routing in Mobile IP is the ability of a mobile node to bypass the home agent when sending IP packets to a destination node. Optional extensions make direct routing possible in Mobile IP for IPv4 (the extensions might not be implemented in all deployments of Mobile IP for IPv4), whereas direct routing is built into Mobile IP for IPv6.

Security

IPSec is a framework of open standards developed by the IETF that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (peers), such as Cisco routers. IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality—The IPSec sender can encrypt packets before sending them across a network.
- Data integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Antireplay—The IPSec receiver can detect and reject replayed packets.



Note

The term *data authentication* is generally used to mean data integrity and data origin authentication. Within this section it also includes antireplay services, unless otherwise specified.

With IPSec, data can be sent across a public network without fear of observation, modification, or spoofing.

IPSec functionality is essentially identical in both IPv6 and IPv4; however, IPSec in IPv6 can be deployed from end-to-end—data may be encrypted along the entire path between a source node and destination node. (Typically, IPSec in IPv4 is deployed between border routers of separate networks.) In IPv6, IPSec is implemented using the authentication extension header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.



Note

Support for IPSec in IPv6 is not in the current Cisco IOS software release. IPSec in IPv6 will be supported in a later release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Transition Richness

Integration and coexistence with IPv4 is a prerequisite to enable the smooth transition of your network to IPv6. Following are some of the techniques available to facilitate the integration of IPv6 networks with IPv4 networks:

- Nodes that support both the IPv4 and IPv6 protocol stacks
- IPv6 tunnels over IPv4 core networks
- Translation gateways (for example, Network Address Translation-Protocol Translation [NAT-PT])

- IPv6 services integration on Multiprotocol Label Switching (MPLS) backbones
- Dedicated IPv6 networks (which support both the IPv6 and IPv4 protocol stacks) over common Layer 2 infrastructures such as Frame Relay, ATM, and optical fiber (for example, Wave-division multiplexing [WDM])

Each technique addresses a different set of attributes that are specific to one context or another. For example, supporting both IPv4 and IPv6 protocol stacks enables nodes to send and receive data on both IPv4 and IPv6 networks. Tunneling encapsulates IPv6 packets in IPv4 packets and uses the IPv4 network as a link-layer mechanism. Other transition techniques address having IPv4-only nodes exchange data with IPv6-only nodes. Additionally, techniques can be combined to achieve a smooth transition of your network to IPv6.



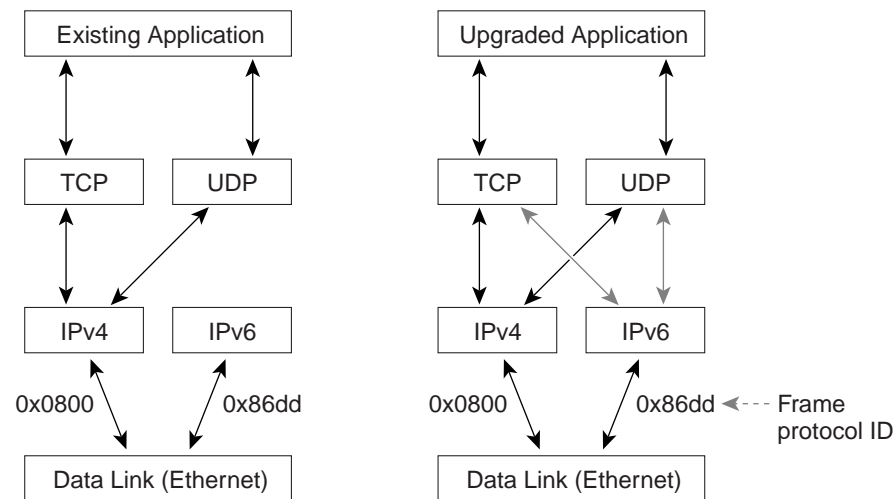
Note

Dual IPv4 and IPv6 protocol stacks and IPv6 tunnels are supported in the current Cisco IOS software release as the primary IPv6 transition techniques. Dedicated IPv6 networks over Layer 2 infrastructures are also supported. NAT-PT and IPv6 services on MPLS backbones will be supported in a later release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Dual IPv4 and IPv6 Protocol Stacks

The preferred technique for a transition to IPv6, the dual IPv4 and IPv6 protocol stack technique enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded—they support only the IPv4 protocol stack—can coexist with upgraded applications on the same node. New and upgraded applications simply make use of both the IPv4 and IPv6 protocol stacks. (See [Figure 20](#).)

Figure 20 Dual IPv4 and IPv6 Protocol Stack Technique

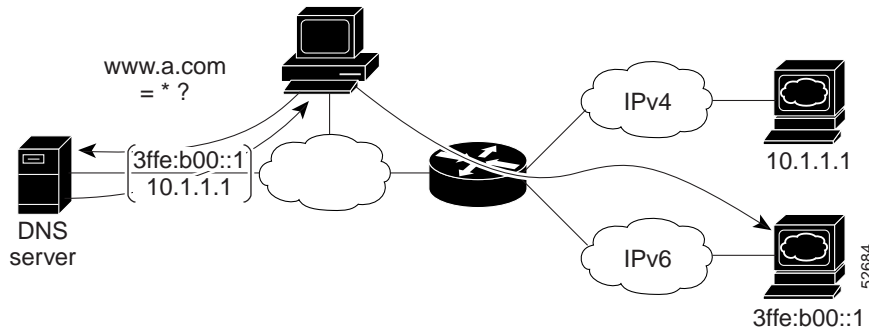


52883

A new application programming interface (API) has been defined to support both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco IOS software supports the dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will forward both IPv4 and IPv6 traffic.

In [Figure 21](#), an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination host name `www.a.com` from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for `www.a.com`. The application chooses an address—in most cases, IPv6 addresses are the default choice—and connects the source node to the destination using the IPv6 protocol stack.

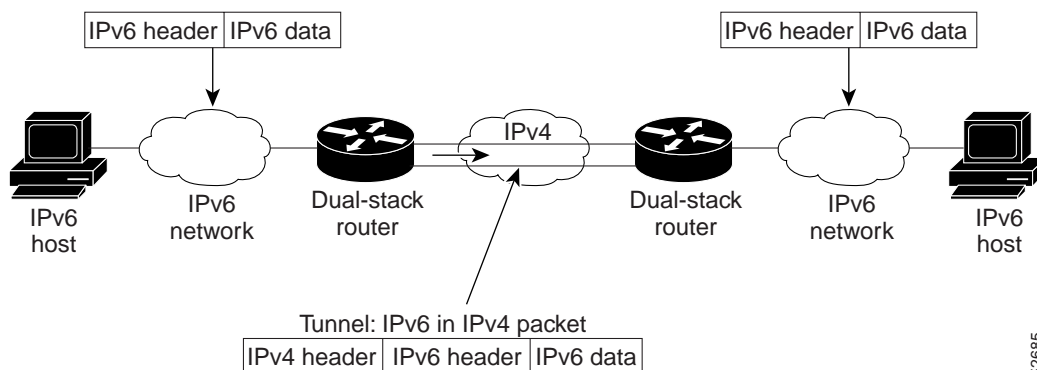
Figure 21 Dual IPv4 and IPv6 Protocol Stack Applications



Overlay Tunnels

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet). (See [Figure 22](#).) By using overlay tunnels, isolated IPv6 networks can communicate without needing to upgrade the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between a border router and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks.

Figure 22 Overlay Tunnels



Note

Overlay tunnels reduce the MTU of an interface by 20 octets (assuming the basic IPv4 packet header does not contain optional fields). A network using overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels connecting isolated IPv6 networks should not be viewed as a final IPv6 network architecture. The use of overlay tunnels should be viewed as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Following are the different IPv6 overlay tunnel types:

- **Manually configured tunnels**—A manually configured IPv6 address is configured on a tunnel interface and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host.
- **Automatic tunnels**—The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an automatic tunnel must support both the IPv4 and IPv6 protocol stacks. Automatic tunnels can be configured between border routers or between a border router and a host.
- **6to4 tunnels**—A 6to4 tunnel is an automatic IPv6 tunnel where a border router in an isolated IPv6 network creates a tunnel to a border router in another isolated IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router that is concatenated to the prefix 2002::/16, in the format 2002:IPv4 address of the border router::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers.

Refer to the *IPv6 for Cisco IOS Software, File 2 of 3: Configuring* and *IPv6 for Cisco IOS Software, File 3 of 3: Commands* documents in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for configuration and command reference information on manually configured tunnels, automatic tunnels, and 6to4 tunnels. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Glossary

6to4 tunnel—An IPv6 automatic tunneling technique where the tunnel endpoint is determined by the globally unique IPv4 address embedded in a 6to4 address. A 6to4 address is a combination of the prefix 2002::/16 and a globally unique 32-bit IPv4 address. (IPv4-compatible addresses are not used in 6to4 tunneling.)

6to4 relay—A 6to4 border router that offers traffic forwarding to the IPv6 internet for other 6to4 border routers. A 6to4 relay forwards packets to any destination that has a 2002::/16 prefix.

aggregatable global unicast address—An IPv6 address that enables strict aggregation of routing prefixes in order to limit the number of routing table entries in the global routing table. Aggregatable global unicast addresses have a fixed prefix of 2000::/3 (001). See also anycast address, IPv6 multicast address, link-local address, and site-local address.

anycast address—An identifier for a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. See also aggregatable global unicast address, IPv6 multicast address, link-local address, site-local address, and solicited-node multicast address.

automatic IPv6 tunnel—An IPv6 tunneling technique where the tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an IPv6 automatic tunnel must support both the IPv4 and IPv6 protocol stacks. Automatic tunnels can be configured between border routers or between a border router and a host. See also IPv4-compatible IPv6 address and manually configured IPv6 tunnel.

CEF—Cisco Express Forwarding. A switching mechanism with performance equivalent to fast switching that scales to support Internet backbone requirements. CEF uses a forwarding information base (FIB) and adjacency table.

CEFv4—Cisco Express Forwarding for IPv4.

CEFv6—Cisco Express Forwarding for IPv6.

dCEFv4—distributed CEF for IPv4.

dCEFv6—distributed CEF for IPv6. Designed for distributed architecture platforms such as the Cisco 12000 series Internet routers.

FIB—forwarding information base (common ISO usage). Database of information used to make forwarding decisions. It is conceptually similar to a routing table or route cache, but very different in implementation.

IPv4-compatible IPv6 address—An IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node, and the IPv4 address embedded in low-order 32-bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks, and are used in automatic tunneling. See also anycast address, automatic IPv6 tunnel, IPv6 multicast address, link-local address, and site-local address.

IPv6 multicast address—An IPv6 address with a prefix of FF00::/8. An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. See also aggregatable global unicast address, anycast address, link-local address, site-local address, and solicited-node multicast address.

link—In IPv6 networks, a network sharing a particular local-link prefix. Links are networks arbitrarily segmented by a network administrator in order to provide a multilevel, hierarchical routing structure while shielding the subnetwork from the addressing complexity of attached networks. Similar to a subnetwork in IPv4. A subnetwork prefix is associated with one link; multiple subnetwork prefixes may be assigned to the same link.

link-local address—An IPv6 unicast address that has a scope limited to the local link (local network). Link-local addresses are automatically configured on all interfaces by using a specific prefix for link-local addresses (FE80::/10) and adding the interface identifier in the modified EUI-64 format. Link-local addresses are used by the neighbor discovery protocol and the router discovery protocol. They also are used by many routing protocols. Link-local addresses can serve as a way to connect devices on the same local network without needing global addresses. See also aggregatable global unicast address, anycast address, IPv6 multicast address, site-local address, and solicited-node multicast address.

manually configured IPv6 tunnel—An IPv6 tunneling technique where a manually configured IPv6 address is configured on a tunnel interface and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host. See also automatic IPv6 tunnel.

RP—Route Processor. Processor module in the Cisco 7000 series routers that contains the CPU, system software, and most of the memory components that are used in the router. Sometimes called a *supervisory processor*.

site-local address—The site-local address is useful only in the context of the site. Its scope is limited to this context. When configured, a site-local address uses a specific prefix (FEC0::/10) and concatenates the subnet ID as a 16-bit field and then the interface identifier in the modified EUI-64 format. See also aggregatable global unicast address, anycast address, IPv6 multicast address, link-local address, and solicited-node multicast address.

solicited-node multicast address—An IPv6 address that has the prefix FF02:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address. The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. Solicited-node multicast addresses are used in neighbor solicitation messages. See also aggregatable global unicast address, anycast address, IPv6 multicast address, link-local address, and site-local address.