IPv6 Configuration Guide

First Published: 2012-08-01

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Read Me First

Feature Information

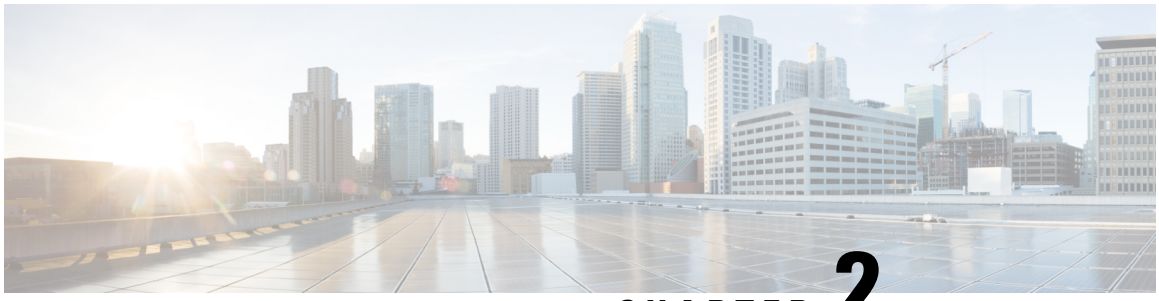
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.



CHAPTER 2

Implementing IPv6 Addressing and Basic Connectivity

Implementing basic IPv6 connectivity in the Cisco IOS software consists of assigning IPv6 addresses to individual router interfaces. The forwarding of IPv6 traffic can be enabled globally, and Cisco Express Forwarding switching for IPv6 can also be enabled. Basic connectivity can be enhanced by configuring support for AAAA record types in the Domain Name System (DNS) name-to-address and address-to-name lookup processes, and by managing IPv6 neighbor discovery.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Toland the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Implementing IPv6 Addressing and Basic Connectivity

- The following prerequisites apply to Cisco Express Forwarding and distributed Cisco Express Forwarding for IPv6:

- To forward IPv6 traffic using Cisco Express Forwarding or distributed Cisco Express Forwarding, you must configure forwarding of IPv6 unicast datagrams globally on the router by using the **ipv6 unicast-routing** command, and you must configure an IPv6 address on an interface by using the **ipv6 address** command.
- You must enable Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef** command before enabling Cisco Express Forwarding for IPv6 globally on the router by using the **ipv6 cef** command.
- On distributed architecture platforms that support both Cisco Express Forwarding and distributed Cisco Express Forwarding, you must enable distributed Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef distributed** command before enabling distributed Cisco Express Forwarding for IPv6 globally on the router by using the **ipv6 cef distributed** command.
- To use Unicast Reverse Path Forwarding (RPF), enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the router. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the router, individual interfaces can be configured with other switching modes.

Note For Unicast RPF to work, Cisco Express Forwarding must be configured globally in the router. Unicast RPF will not work without Cisco Express Forwarding.

Restrictions for Implementing IPv6 Addressing and Basic Connectivity

- Multiple IPv6 global addresses within the same prefix can be configured on an interface; however, multiple IPv6 link-local addresses on an interface are not supported.
- IPv4 alias and IPv6 alias addresses used must be available in the global routing table and not under VRF.

Information About Implementing IPv6 Addressing and Basic Connectivity

IPv6 for Cisco Software

IPv6, formerly named IPng (next generation), is the latest version of the Internet Protocol (IP). IP is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IPv6 was proposed when it became clear that the 32-bit addressing scheme of IP version 4 (IPv4) was inadequate to meet the demands of Internet growth. After extensive discussion it was decided to base IPng on IP but add a much larger address space and improvements such as a simplified main header and extension headers. IPv6 is described initially in RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, issued by the Internet Engineering Task Force (IETF). Further RFCs describe the architecture and services supported by IPv6.

The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6 while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses. The

larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. IPv6 prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities provide an IPv6 addressing hierarchy that allows for more efficient routing. IPv6 supports widely deployed routing protocols such as Routing Information Protocol (RIP), Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF) for IPv6, and multiprotocol Border Gateway Protocol (BGP). Other available features include stateless autoconfiguration and an increased number of multicast addresses.

Large IPv6 Address Space for Unique Addresses

The primary motivation for IPv6 is the need to meet 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; therefore, IPv6 enables new application protocols that do not require special processing by border devices 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:x. Following are two examples of IPv6 addresses:

2001:DB8:7654:3210:FEDC:BA98:7654:3210

2001:DB8:0:0:8:800:200C:417A

IPv6 addresses commonly contain successive hexadecimal fields of zeros. 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). The table below 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 be used only once in an IPv6 address to represent the longest successive hexadecimal fields of zeros. The hexadecimal letters in IPv6 addresses are not case-sensitive.

Table 1: Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:DB8:800:200C:417A	2001::DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

The loopback address listed in the table above 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 devices do not forward packets that have the IPv6 loopback address as their source or destination address.

The unspecified address listed in the table above 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. The unspecified IPv6 addresses must 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* 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 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, 2001:DB8:8086:6502::/32 is a valid IPv6 prefix.

IPv6 Address Type: Unicast

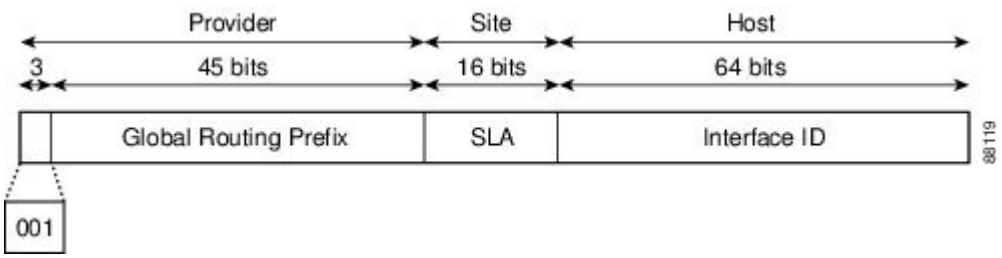
An IPv6 unicast address is an identifier for a single interface, on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address. Cisco software supports the IPv6 unicast address types described in the following sections.

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 are used on links that are aggregated upward through organizations, and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). The figure below shows the structure of an aggregatable global address.

Figure 1: Aggregatable Global Address Format



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address typically consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields named Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLS and NLA fields from the RFCs because these fields are policy-based. Some existing IPv6 networks deployed before the change might still be using networks based on the older architecture.

A 16-bit subnet field called the subnet ID could be used by individual organizations to create their own local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

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

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

- For all IEEE 802 interface types (for example, FDDI interfaces), the first three octets (24 bits) are taken from the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (the Media Access Control [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 ID 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 interface types--except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is constructed in the same way as the interface ID 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 (because the interface does not have a MAC address).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface ID 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 Point-to-Point Protocol (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. The first MAC address in the router is used to construct the identifier for interfaces using PPP.

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.

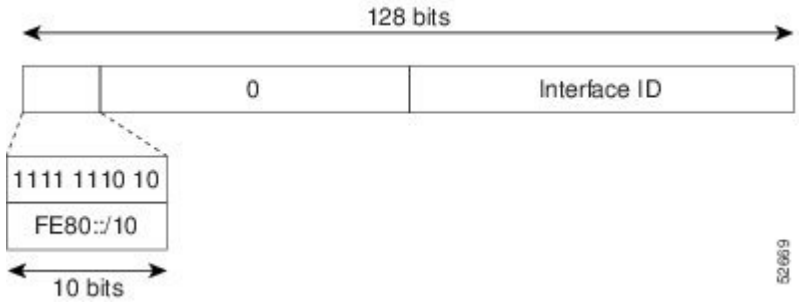
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a message digest algorithm 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

Link-Local Address

A link-local address is an IPv6 unicast address that can be automatically configured on any interface 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 globally unique addresses to communicate. The figure below shows the structure of a link-local address.

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

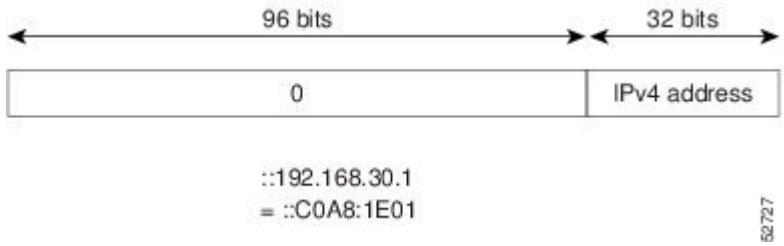
Figure 2: Link-Local Address Format



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 0:0:0:0:0:0: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. The figure below shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 3: IPv4-Compatible IPv6 Address Format

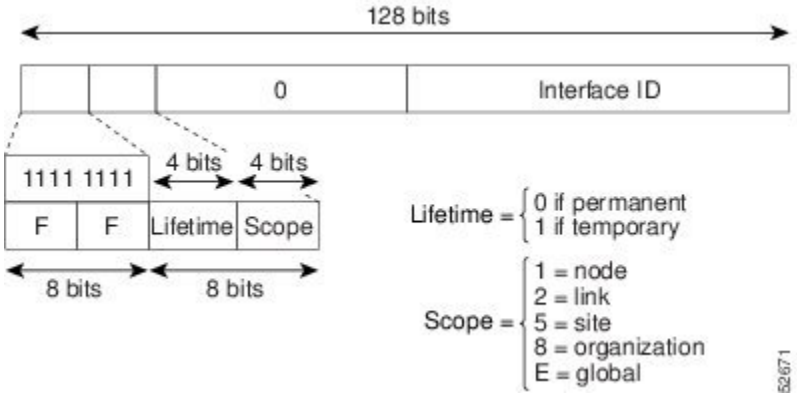


IPv6 Address Type Multicast

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. The figure below shows the format of the IPv6 multicast address.

Figure 4: IPv6 Multicast Address Format



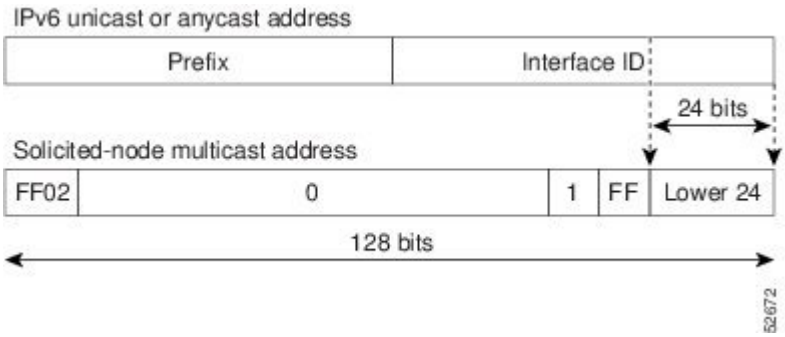
An IPv6 address must be configured on an interface for the interface to forward IPv6 traffic. Configuring a global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2

Note The solicited-node multicast address is used in the Neighbor Discovery process.

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast address (see the figure below). 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 5: IPv6 Solicited-Node Multicast Address Format



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

IPv6 Address Output Display

When IPv6 or IPv4 command output displays an IPv6 address, a long IPv6 address can overflow into neighboring fields, causing the output to be difficult to read. The output fields were designed to work with the longest possible IPv4 address, which has 15 characters; IPv6 addresses can be up to 39 characters long. The following scheme has been adopted in IPv4 and IPv6 commands to allow the appropriate length of IPv6 address to be displayed and move the following fields to the next line, if necessary. The fields that are moved are kept in alignment with the header row.

The following example displays eight connections. The first six connections feature IPv6 addresses; the last two connections feature IPv4 addresses.

```
Device# where
Conn Host                Address             Byte  Idle Conn Name
   1 test5               2001:DB8:3333:4::5    6    24 test5
   2 test4               2001:DB8:3333:44::5
                                               6    24 test4
   3 2001:DB8:3333:4::5 2001:DB8:3333:4::5     6    24 2001:DB8:3333:4::5
   4 2001:DB8:3333:44::5
                         2001:DB8:3333:44::5
                                               6    23 2001:DB8:3333:44::5
   5 2001:DB8:3000:4000:5000:6000:7000:8001
                         2001:DB8:3000:4000:5000:6000:7000:8001
                                               6    20 2001:DB8:3000:4000:5000:6000:
   6 2001:DB8:1::1      2001:DB8:1::1          0     1 2001:DB8:1::1
   7 10.1.9.1            10.1.9.1              0     0 10.1.9.1
   8 10.222.111.222      10.222.111.222        0     0 10.222.111.222
```

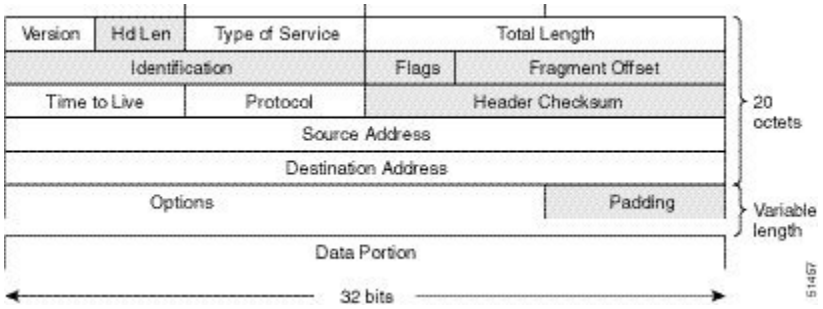
Connection 1 contains an IPv6 address that uses the maximum address length in the address field. Connection 2 shows the IPv6 address overflowing the address field and the following fields moved to the next line, but in alignment with the appropriate headers. Connection 3 contains an IPv6 address that fills the maximum length of the hostname and address fields without wrapping any lines. Connection 4 shows the effect of both the hostname and address fields containing a long IPv6 address. The output is shown over three lines keeping the correct heading alignment. Connection 5 displays a similar effect as connection 4 with a very long IPv6 address in the hostname and address fields. Note that the connection name field is actually truncated. Connection 6 displays a very short IPv6 address that does not require any change in the display. Connections 7 and 8 display short and long IPv4 addresses.

Note The IPv6 address output display applies to all commands that display IPv6 addresses.

Simplified IPv6 Packet Header

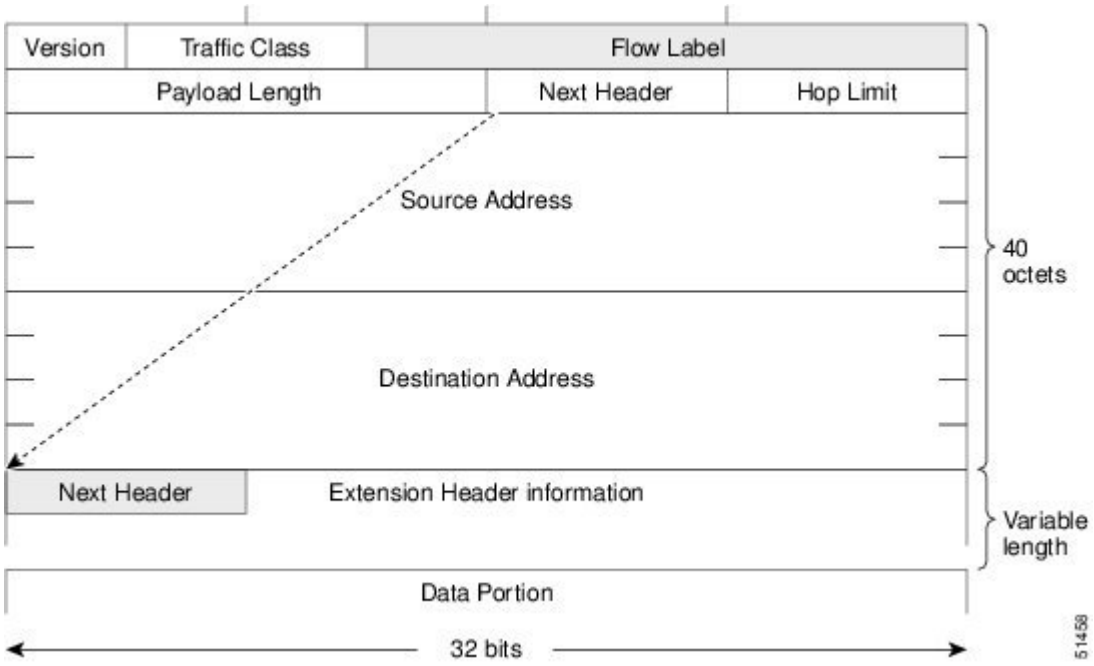
The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see the figure below). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size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

Figure 6: IPv4 Packet Header Format



The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see the figure below). Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by devices 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 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 7: IPv6 Packet Header Format



The table below lists the fields in the basic IPv6 packet header.

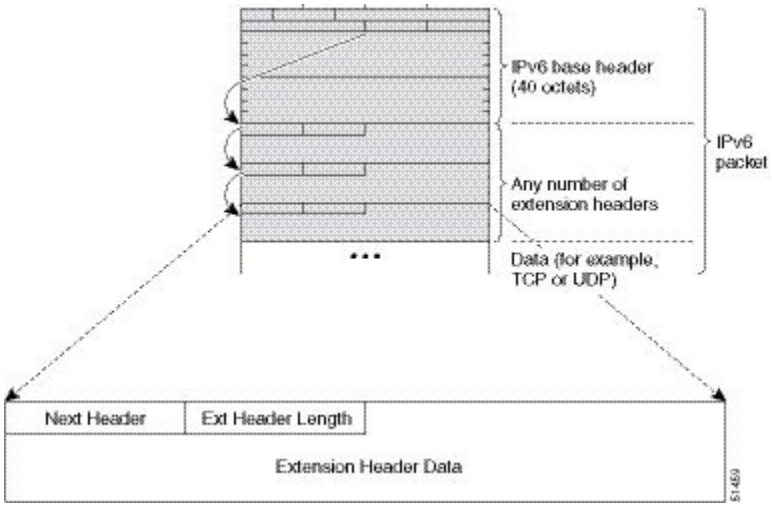
Table 2: Basic IPv6 Packet Header Fields

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 the figure immediately above.

Field	Description
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 devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the value by one. Because no checksum is in the IPv6 header, the device 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. 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. The figure below shows the IPv6 extension header format.

Figure 8: IPv6 Extension Header Format



The table below 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.

Header Type	Next Header Value	Description
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.
Authentication header and ESP header	51 50	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 headers	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.
Mobility headers	135	Extension headers used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings.

Cisco Express Forwarding for IPv6

Cisco Express Forwarding is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets.

Each IPv6 router interface has an association to one IPv6 global FIB and one IPv6 link-local FIB (multiple interfaces can have an association to the same FIB). All IPv6 router interfaces that are attached to the same IPv6 link share the same IPv6 link-local FIB. IPv6 packets that have an IPv6 global destination address are processed by the IPv6 global FIB; however, packets that have an IPv6 global destination address and an IPv6 link-local source address are sent to the RP for process switching and scope-error handling. Packets that have a link-local source address are not forwarded off of the local link and are sent to the RP for process switching and scope-error handling.

Unicast Reverse Path Forwarding

Use the Unicast Reverse Path Forwarding for IPv6 feature to mitigate problems caused by malformed or spoofed IPv6 source addresses that pass through an IPv6 device. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IPv6 address spoofing.

When uRPF is enabled on an interface, the device examines all packets received on that interface. The device verifies that the source address appears in the routing table and matches the interface on which the packet was received. This "look backward" ability is available only when Cisco Express Forwarding is enabled on the device; this is because the lookup relies on the presence of the Forwarding Information Bases (FIBs). Cisco Express Forwarding generates the FIB as part of its operation.

Note uRPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

Note With uRPF, all equal-cost "best" return paths are considered valid. uRPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses. IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

The table below lists the IPv6 DNS record types.

Table 4: IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.) **Note** Cisco software supports resolution of PTR records for the IP6.INT domain.	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

Cisco Discovery Protocol IPv6 Address Support

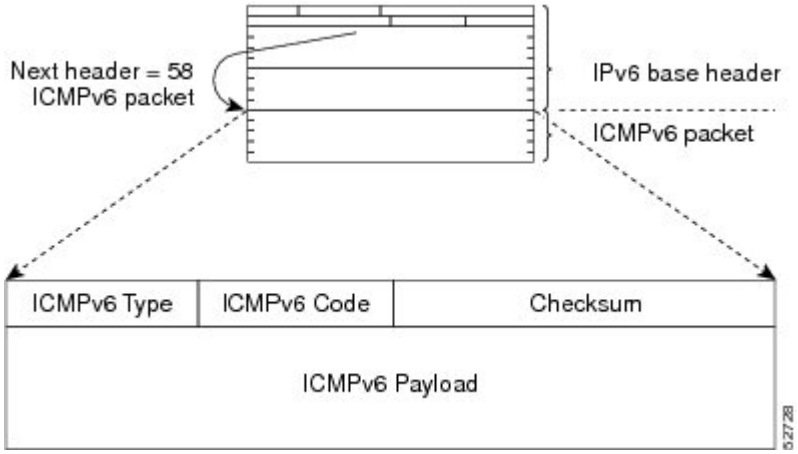
The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

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, such as 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 devices 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 follows 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. The figure below shows the IPv6 ICMP packet header format.

Figure 9: IPv6 ICMP Packet Header Format



IPv6 ICMP Rate Limiting

The IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The initial implementation of IPv6 ICMP rate limiting defined a fixed interval between error messages, but some applications such as traceroute often require replies to a group of requests sent in rapid succession. The fixed interval between error messages is not flexible enough to work with applications such as traceroute and can cause the application to fail.

Implementing a token bucket scheme allows a number of tokens--representing the ability to send one error message each--to be stored in a virtual bucket. The maximum number of tokens allowed in the bucket can be specified, and for every error message to be sent, one token is removed from the bucket. If a series of error messages is generated, error messages can be sent until the bucket is empty. When the bucket is empty of tokens, no IPv6 ICMP error messages are sent until a new token is placed in the bucket. The token bucket algorithm does not increase the average rate limiting time interval, and it is more flexible than the fixed time interval scheme.

IPv6 MTU Path 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 one 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 device processing resources and helps IPv6 networks run more efficiently.

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

With IPv6 path MTU discovery, a device originating IPv6 traffic has an MTU cache that contains MTU values received in ICMPv6 "toobig" messages. In order to prevent an attacker from filling the MTU cache, the device keeps track of the destinations to which it has originated (sent) traffic, and only accepts toobig ICMPv6 messages that have an inner destination matching one of these tracked destinations.

If a malicious device can learn to which destination the device is originating traffic, it could still send a toobig ICMPv6 message to the device for this destination, even if the attacker is not on the path to this destination, and succeeds in forcing his entry into the MTU cache. The device then starts fragmenting traffic to this destination, which significantly affects device performance.

Enabling flow-label marking for locally generated traffic can mitigate this attack. Originated packets are marked with a flow label (which is randomly generated and changed every minute), and toobig messages received are checked against the values sent. Unless an attacker can snoop traffic, the attacker will not know which flow label to use, and its toobig message will be dropped.

IPv6 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 track neighboring devices.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Stateful Switchover

IPv6 neighbor discovery supports stateful switchover (SSO) using Cisco Express Forwarding. When switchover occurs, the Cisco Express Forwarding adjacency state, which is checkpointed, is used to reconstruct the neighbor discovery cache.

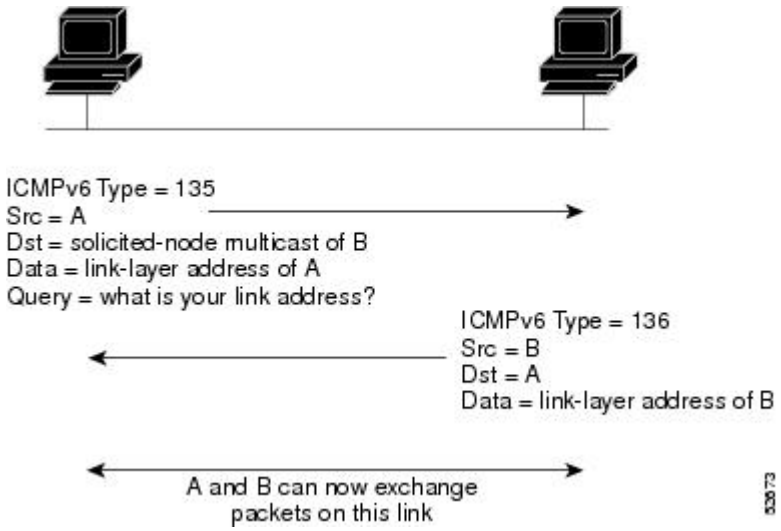
SSO and ISSU Support for Per-User IPv6 ACL for PPP Sessions

The Stateful Switchover (SSO) and In Service Software Upgrade (ISSU) support for per-user IPv6 ACL for PPP sessions feature reproduces IPv6 ACLs on the active RP to the standby RP, which provide a consistent SSO and ISSU experience for active sessions.

IPv6 Neighbor Solicitation Message

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 the figure below). 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 10: 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 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 advertisement 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 devices). 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 device 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.

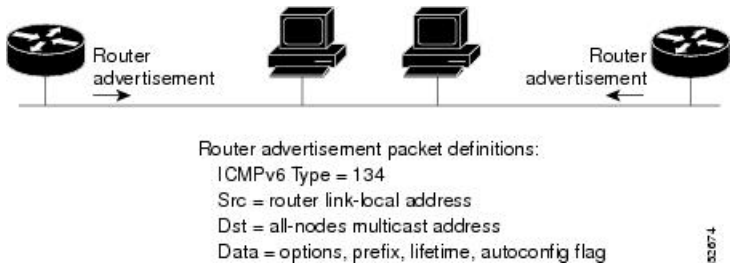
Every IPv6 unicast address (global or link-local) must be verified for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address. The Cisco implementation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

IPv6 Router Advertisement Message

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

The RA messages are sent to the all-nodes multicast address (see the figure below).

Figure 11: IPv6 Neighbor Discovery--RA Message



RA 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 stateful) 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

RAs are also sent in response to router solicitation messages.

The following RA message parameters can be configured:

- The time interval between periodic RA 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 RA messages (with default values) is automatically enabled on FDDI interfaces when the **ipv6 unicast-routing** command is configured. For other interface types, the sending of RA messages must be manually configured by using the **no ipv6 nd ra suppress**command. The sending of RA messages can be disabled on individual interfaces by using the **ipv6 nd ra**suppress command.

Default Router Preferences for Traffic Engineering

Hosts discover and select default devices by listening to router advertisements (RAs). Typical default device selection mechanisms are suboptimal in certain cases, such as when traffic engineering is needed. For example, two devices on a link may provide equivalent but not equal-cost routing, and policy may dictate that one of the devices is preferred. Some examples are as follows:

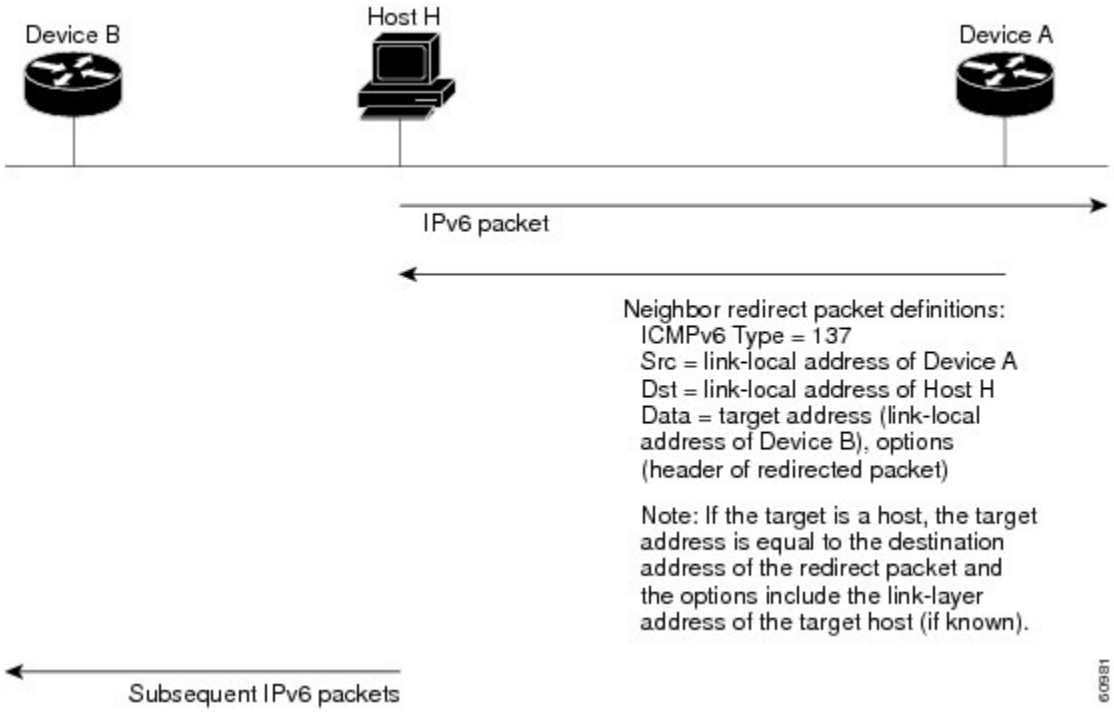
- Multiple devices that route to distinct sets of prefixes--Redirects (sent by nonoptimal devices for a destination) mean that hosts can choose any device and the system will work. However, traffic patterns may mean that choosing one of the devices would lead to considerably fewer redirects.

- Accidentally deploying a new device--Deploying a new device before it has been fully configured could lead to hosts adopting the new device as a default device and traffic disappearing. Network managers may want to indicate that some devices are more preferred than others.
- Multihomed situations--Multihomed situations may become more common, because of multiple physical links and because of the use of tunneling for IPv6 transport. Some of the devices may not provide full default routing because they route only to the 6-to-4 prefix or they route only to a corporate intranet. These situations cannot be resolved with redirects, which operate only over a single link.

IPv6 Neighbor Redirect Message

A value of 137 in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Devices send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see the figure below).

Figure 12: IPv6 Neighbor Discovery: Neighbor Redirect Message



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

After forwarding a packet, a device 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 device.
- The packet is about to be sent out the interface on which it was received.
- The device 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 device generates all IPv6 ICMP error messages, including neighbor redirect messages, which ultimately reduces link-layer congestion.

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

Per-Interface Neighbor Discovery Cache Limit

The number of entries in the Neighbor Discovery cache can be limited by interface. Once the limit is reached, no new entries are allowed. The per-interface Neighbor Discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the Neighbor Discovery cache, whether intentionally or unintentionally.

When this feature is enabled globally, a common per-interface cache size limit is configured on all interfaces on the device. When this feature is enabled per interface, a cache size limit is configured on the associated interface. The per-interface limit overrides any globally configured limit.

Link, Subnet, and Site Addressing Changes

This section describes the IPv6 stateless autoconfiguration and general prefix features, which can be used to manage link, subnet, and site addressing changes.

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::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

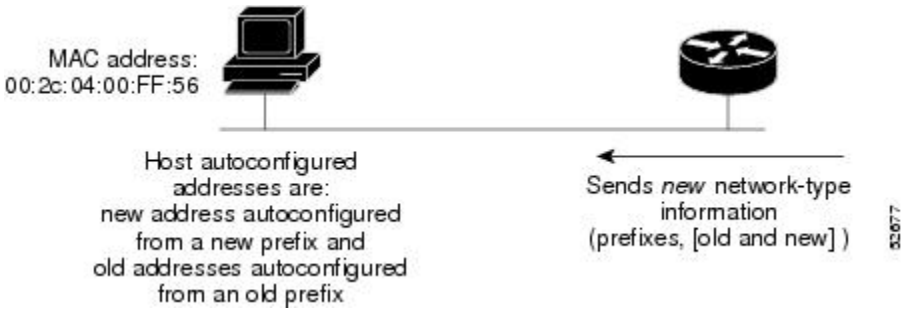
Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) server. With IPv6, a device on the link advertises any global prefixes in Router Advertisement (RA) messages, as well as its willingness to function as a default device for the link. RA messages are sent periodically and in response to device solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA 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 RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Device 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 RA 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 RA messages that are sent on the link. (The RA 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. Configuration of the lifetime parameters associated with the old and new prefixes means that 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 RA messages and only addresses that contain the new prefix are used on the link (the renumbering is complete) (see the figure below).

Figure 13: IPv6 Network Renumbering for Hosts Using Stateless Autoconfiguration



IPv6 General Prefixes

The upper 64 bits of an IPv6 address are composed from a global routing prefix plus a subnet ID, as defined in RFC 3513. A general prefix (for example, /48) holds a short prefix, based on which a number of longer, more-specific prefixes (for example, /64) can be defined. When the general prefix is changed, all of the more-specific prefixes based on it will change, too. This function greatly simplifies network renumbering and allows for automated prefix definition.

For example, a general prefix might be 48 bits long (“/48”) and the more specific prefixes generated from it might be 64 bits long (“/64”). In the following example, the leftmost 48 bits of all the specific prefixes will be the same, and they are the same as the general prefix itself. The next 16 bits are all different.

```
 General prefix: 2001:DB8:2222::/48
Specific prefix: 2001:DB8:2222:0000::/64
Specific prefix: 2001:DB8:2222:0001::/64
Specific prefix: 2001:DB8:2222:4321::/64
Specific prefix: 2001:DB8:2222:7744::/64
```

General prefixes can be defined in several ways:

- Manually
- Based on a 6to4 interface
- Dynamically, from a prefix received by a Dynamic Host Configuration Protocol (DHCP) for IPv6 prefix delegation client

More specific prefixes, based on a general prefix, can be used when configuring IPv6 on an interface.

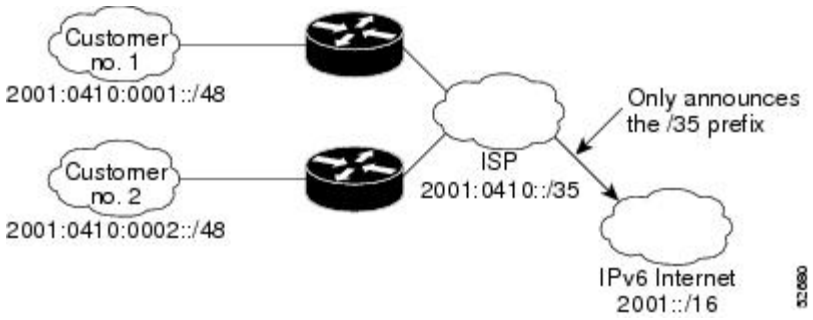
DHCP for IPv6 Prefix Delegation

DHCP for IPv6 can be used in environments to deliver stateful and stateless information. For further information about this feature, see *Implementing DHCP for IPv6*.

IPv6 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 the figure below).

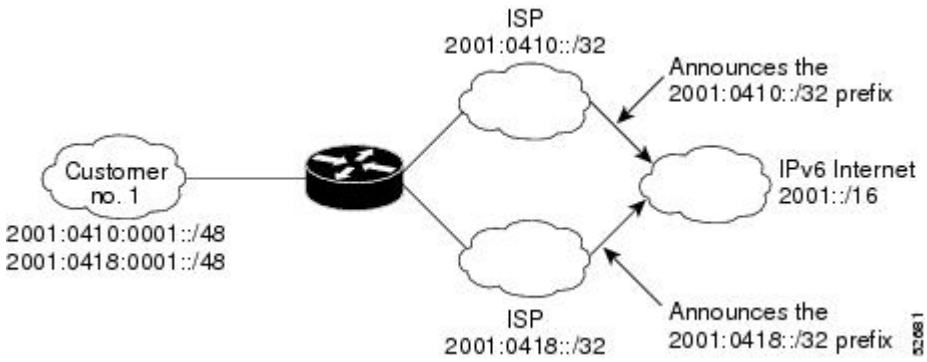
Figure 14: IPv6 Prefix Aggregation



IPv6 Site Multihoming

Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network allows that network to connect easily to multiple ISPs without breaking the global routing table (see the figure below).

Figure 15: IPv6 Site Multihoming



IPv6 Data Links

In IPv6 networks, a data link is a network sharing a particular link-local prefix. Data 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. The function of a

subnetwork in IPv6 is similar to a subnetwork in IPv4. A subnetwork prefix is associated with one data link; multiple subnetwork prefixes may be assigned to the same data link.

The following data links are supported for IPv6: FDDI, Frame Relay PVC, Cisco High-Level Data Link Control (HDLC), PPP over Packet over SONET, ISDN, and serial interfaces.

How to Implement IPv6 Addressing and Basic Connectivity

Configuring IPv6 Addressing and Enabling IPv6 Routing

Perform this task to assign IPv6 addresses to individual device interfaces and enable IPv6 traffic forwarding globally on the device. By default, IPv6 addresses are not configured and IPv6 routing is disabled.

Note Multiple IPv6 link-local addresses on an interface are not supported.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ipv6 address** *ipv6-prefix* /*prefix-length* **eui-64**
   -
   - **ipv6 address** *ipv6-address* / *prefix-length* **link-local**
   -
   -
   - **ipv6 enable**
5. **exit**
6. **ipv6 unicast-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.

Step 4	Do one of the following:
	• **ipv6 address** *ipv6-prefix /prefix-length* **eui-64**
	•
	• **ipv6 address** *ipv6-address / prefix-length* **link-local**
	•
	•
	• **ipv6 enable**
	Example:
	`Device(config-if)# ipv6 address 2001:DB8:0:1::/64 eui-64`
	Example:
	Example:
	`Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local`
	Example:
	Example:
	Example:
	`Device(config-if)# ipv6 enable`
		Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface.
		or
		Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.
		or
		Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link.
		• Specifying the **ipv6 address eui-64** command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID.
		• Specifying the **ipv6 address link-local** command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.

Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode, and returns the device to global configuration mode.
Step 6	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

Configuring a Neighbor Discovery Cache Limit

Configuring a Neighbor Discovery Cache Limit on a Specified Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config-if)# ipv6 nd cache interface-limit 1`	Configures a Neighbor Discovery cache limit on a specified interface on the device. • Issuing this command overrides any configuration that may have been created by issuing the **ipv6 nd cache interface-limit** in global configuration mode.

Configuring a Neighbor Discovery Cache Limit on All Device Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config)# ipv6 nd cache interface-limit 4`	Configures a neighbor discovery cache limit on all interfaces on the device.

Defining and Using IPv6 General Prefixes

General prefixes can be defined in several ways:

- Manually
- Based on a 6to4 interface
- Dynamically, from a prefix received by a DHCP for IPv6 prefix delegation client

More specific prefixes, based on a general prefix, can be used when configuring IPv6 on an interface.

Defining a General Prefix Manually

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 general-prefix** *prefix-name* {*ipv6-prefix/prefix-length* | **6to4** *interface-type interface-number*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 general-prefix** *prefix-name* {*ipv6-prefix/prefix-length* | **6to4** *interface-type interface-number*} **Example:** `Device(config)# ipv6 general-prefix my-prefix 2001:DB8:2222::/48`	Defines a general prefix for an IPv6 address.

Defining a General Prefix Based on a 6to4 Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 general-prefix** *prefix-name* {*ipv6-prefix / prefix-length* | **6to4** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 general-prefix** *prefix-name* {*ipv6-prefix / prefix-length* | **6to4** *interface-type interface-number* **Example:** `Router(config)# ipv6 general-prefix my-prefix 6to4 gigabitethernet 0/0/0`	Defines a general prefix for an IPv6 address. When defining a general prefix based on a 6to4 interface, specify the **6to4** keyword and the *interface-type interface-number*arguments. When defining a general prefix based on an interface used for 6to4 tunneling, the general prefix will be of the form 2001:a.b.c.d::/48, where "a.b.c.d" is the IPv4 address of the interface referenced.

Defining a General Prefix with the DHCP for IPv6 Prefix Delegation Client Function

You can define a general prefix dynamically using the DHCP for IPv6 prefix delegation client function. For information on how to perform this task, see the Implementing DHCP for IPv6 module.

Using a General Prefix in IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length* **Example:** `Router(config-if) ipv6 address my-prefix 2001:DB8:0:7272::/64`	Configures an IPv6 prefix name for an IPv6 address and enables IPv6 processing on the interface.

Customizing IPv6 ICMP Rate Limiting

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 icmp error-interval** *milliseconds* [*bucketsize*] **Example:** `Device(config)# ipv6 icmp error-interval 50 20`	Customizes the interval and bucket size for IPv6 ICMP error messages.

Enabling Flow-Label Marking in Packets that Originate from the Device

This feature allows the device to track destinations to which the device has sent packets that are 1280 bytes or larger.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flowset**
4. **exit**
5. **clear ipv6 mtu**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 flowset** **Example:** `Device(config)# ipv6 flowset`	Configures flow-label marking in 1280-byte or larger packets sent by the device.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode, and places the device in privileged EXEC mode.
Step 5	**clear ipv6 mtu** **Example:** `Device# clear ipv6 mtu`	Clears the MTU cache of messages.

Clearing Messages from the IPv6 MTU Cache

SUMMARY STEPS

1. **enable**
2. **clear ipv6 mtu**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted
Step 2	**clear ipv6 mtu** **Example:** `Device# clear ipv6 mtu`	Clears the MTU cache of messages.

Configuring the DRP Extension for Traffic Engineering

Perform this task to configure the DRP extension to RAs in order to signal the preference value of a default router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd router-preference** {**high** | **medium** | **low**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 nd router-preference** {**high** | **medium** | **low** **Example:** `Router(config-if)# ipv6 nd router-preference high`	Configures a DRP for a router on a specific interface

Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6

Configuring Cisco Express Forwarding

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do the following:
   - **ipv6 cef**
4. **ipv6 cef accounting** [**non-recursive** | **per-prefix** | **prefix-length**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	Do the following: • **ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables Cisco Express Forwarding globally on the device.
Step 4	**ipv6 cef accounting** [**non-recursive** | **per-prefix** | **prefix-length**] **Example:** `Device(config)# ipv6 cef accounting`	Enables Cisco Express Forwarding network accounting globally on the device. • Network accounting for Cisco Express Forwarding enables you to better understand Cisco Express Forwarding traffic patterns within your network by collecting statistics specific to Cisco Express Forwarding traffic. For example, network accounting for Cisco Express Forwarding enables you to collect information such as the number of packets and bytes switched to a destination or the number of packets switched through a destination. • The optional **per-prefix** keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 destination (or IPv6 prefix). • The optional **prefix-length** keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 prefix length. **Note** When Cisco Express Forwarding is enabled globally on the device, accounting information is collected at the RP.

Configuring Unicast RPF

Before you begin

To use uRPF, enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the device. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the device, individual interfaces can be configured with other switching modes.

Note Cisco Express Forwarding must be configured globally in the device. uRPF will not work without Cisco Express Forwarding.

Note uRPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, meaning that there are multiple routes to the source of a packet. uRPF should be applied only where there is natural or configured symmetry.

For example, devices at the edge of the network of an ISP are more likely to have symmetrical reverse paths than devices that are in the core of the ISP network. Devices that are in the core of the ISP network have no guarantee that the best forwarding path out of the device will be the path selected for packets returning to the device. Therefore, we do not recommend that you apply uRPF where there is a chance of asymmetric routing. It is simplest to place uRPF only at the edge of a network or, for an ISP, at the customer edge of the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 verify unicast source reachable-via** {**rx** | **any**} [**allow-default**] [**allow-self-ping**] [*access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 verify unicast source reachable-via** {**rx** | **any**} [**allow-default**] [**allow-self-ping**] [*access-list-name* **Example:** `Device(config-if)# ipv6 verify unicast source reachable-via any`	Verifies that a source address exists in the FIB table and enables uRPF.

Mapping Hostnames to IPv6 Addresses

Hostname-to-Address Mappings

A name server is used to track information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IPv4 addresses, IPv6 addresses, or both address types. In order to use this service to map domain names to IPv6 addresses, you must specify a name server and enable the DNS, which is the global naming scheme of the Internet that uniquely identifies network devices.

Cisco software maintains a cache of hostname-to-address mappings for use by the **connect**, **telnet**, and ping commands, related Telnet support operations, and many other commands that generate command output. This cache speeds the conversion of names to addresses.

Similar to IPv4, IPv6 uses a naming scheme that allows a network device to be identified by its location within a hierarchical name space that provides for domains. Domain names are joined with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that is identified by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the FTP server, for example, is identified as *ftp.cisco.com*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip domain name** [**vrf** *vrf-name*] *name*
   -
   -
   - **ip domain lis**t [**vrf** *vrf-name*] *name*
4. **ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*]
5. **ip domain-lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **ip domain name** [**vrf** *vrf-name*] *name* •	(Optional) Defines a default domain name that Cisco software will use to complete unqualified hostnames. or

	Command or Action	Purpose

	•
	• **ip domain lis**t [**vrf** *vrf-name*] *name*
	Example:
	`Device(config)# ip domain-name cisco.com`
	Example:
	Example:
	`Device(config)# ip domain list cisco1.com`
		(Optional) Defines a list of default domain names to complete unqualified hostnames.
		• You can specify a default domain name that Cisco software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		Note The **ip domain name** and **ip domain list** commands are used to specify default domain names that can be used by both IPv4 and IPv6.

Step 4	**ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*] **Example:** `Device(config)# ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1`	Specifies one or more hosts that supply name information. • Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS. **Note** The *server-address*argument can be either an IPv4 or IPv6 address.
Step 5	**ip domain-lookup** **Example:** `Device(config)# ip domain-lookup`	Enables DNS-based address translation. • DNS is enabled by default.

Mapping IPv6 Addresses to IPv6 Frame Relay Interfaces

Perform this task to map IPv6 addresses to Frame Relay PVCs. Specifically, the steps in this section explain how to explicitly map IPv6 addresses to the Frame Relay PVCs used to reach the addresses.

Note This task shows how to configure Frame Relay PVCs. Several of the steps are labeled optional because many networks will require only one type of PVC to be configured.

IPv6 for Cisco IOS XE Software Support for Wide-Area Networking Technologies

IPv6 for Cisco IOS XE software supports wide-area networking technologies such as Cisco HDLC, PPP over Packet over SONET (PoS), ISDN, and serial (synchronous and asynchronous) interface types, and Frame Relay PVCs. These technologies function the same in IPv6 as they do in IPv4--IPv6 does not enhance the technologies in any way.

IPv6 Addresses and PVCs

Broadcast and multicast are used in LANs to map protocol (network-layer) addresses to the hardware addresses of remote nodes (hosts and routers). Because using broadcast and multicast to map network-layer addresses to hardware addresses in circuit-based WANs such as Frame Relay networks is difficult to implement, these networks utilize implicit, explicit, and dynamic mappings for the network-layer addresses of remote nodes and the PVCs used to reach the addresses.

Assigning an IPv6 address to an interface by using the **ipv6 address** command defines the IPv6 addresses for the interface and the network that is directly connected to the interface. If only one PVC is terminated on the interface (the interface is a point-to-point interface), there is an implicit mapping between all of the IPv6 addresses on the network and the PVC used to reach the addresses (no additional address mappings are needed). If several PVCs are terminated on the interface (the interface is a point-to-multipoint interface), the frame-relay map ipv6 command is used to configure explicit mappings between the IPv6 addresses of the remote nodes and the PVCs used to reach the addresses.

Note Given that IPv6 supports multiple address types, and depending on which applications or protocols are configured on a point-to-multipoint interface, you may need to configure multiple explicit mappings between the IPv6 addresses of the interface and the PVC used to reach the addresses. For example, explicitly mapping both the link-local and global IPv6 address of a point-to-multipoint interface to the PVC that the interface terminates ensures that the Interior Gateway Protocol (IGP) configured on the interface forwards traffic to and from the PVC correctly.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **protocol ipv6** *ipv6-address* [[**no**] **broadcast**]
5. **frame-relay map ipv6** *ipv6-address dlci* [**broadcast**] [**cisco**] [**ietf**] [**payload-compression packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}]
6. **ipv6 address** *ipv6-address* / *prefix-length* **link-local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial 3`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**protocol ipv6** *ipv6-address* [[**no**] **broadcast**] **Example:** `Router(config-if-atm-vc)# protocol ipv6 2001:DB8:2222:1003::45`	(Optional) Maps the IPv6 address of a remote node to the PVC used to reach the address.
Step 5	**frame-relay map ipv6** *ipv6-address dlci* [**broadcast**] [**cisco**] [**ietf**] [**payload-compression packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}] **Example:** `Router(config-if)# frame-relay map ipv6 FE80::E0:F727:E400:A 17 broadcast`	(Optional) Maps the IPv6 address of a remote node to the data-link connection identifier (DLCI) of the PVC used to reach the address.
Step 6	**ipv6 address** *ipv6-address / prefix-length* **link-local** **Example:** `Router(config-if)# ipv6 address 2001:DB8:2222:1044::46/64 link-local`	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. • In the context of this task, a link-local address of the node at the other end of the link is required for the IGP used in the network. • Specifying the **ipv6 address link-local** command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.

Displaying IPv6 Redirect Messages

SUMMARY STEPS

1. **enable**
2. **show ipv6 interface** [**brief**] [*type number*] [**prefix**]
3. **show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*]
4. **show ipv6 traffic**
5. **show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**]
6. **enable**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 interface** [**brief**] [*type number*] [**prefix**] **Example:** `Device# show ipv6 interface gigabitethernet 0/0/0`	Displays the usability status of interfaces configured for IPv6.
Step 3	**show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*] **Example:** `Device# show ipv6 route`	(Optional) Displays the current contents of the IPv6 routing table.
Step 4	**show ipv6 traffic** **Example:** `Device# show ipv6 traffic`	(Optional) Displays statistics about IPv6 traffic.
Step 5	**show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**] **Example:** `Device# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
Step 6	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 7	**show running-config** **Example:** `Device# show running-config`	Displays the current configuration running on the device.

Examples

Sample Output from the show ipv6 route Command

When the *ipv6-address* or *ipv6-prefix*/*prefix-length* argument is specified, only route information for that address or network is displayed. The following is sample output from the **show ipv6 route** command when entered with the IPv6 prefix 2001:DB8::/35:

```
Router# show ipv6 route 2001:DB8::/35
IPv6 Routing Table - 261 entries
```

```
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
B 2001:DB8::/35 [20/3]
  via FE80::60:5C59:9E00:16, Tunnel1
```

Sample Output from the show ipv6 traffic Command

In the following example, the **show ipv6 traffic** command is used to display ICMP rate-limited counters:

```
Router# show ipv6 traffic
ICMP statistics:
  Rcvd: 188 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
        unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        1 router solicit, 175 router advert, 0 redirects
        0 neighbor solicit, 12 neighbor advert
  Sent: 7376 output, 56 rate-limited
        unreach: 0 routing, 15 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        15 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        0 router solicit, 7326 router advert, 0 redirects
        2 neighbor solicit, 22 neighbor advert
```

Sample Output from the show frame-relay map Command

In the following example, the show frame-relay map command is used to verify that the IPv6 address of a remote node is mapped to the DLCI of the PVC used to reach the address. The following example shows that the link-local and global IPv6 addresses (FE80::E0:F727:E400:A and 2001:DB8:2222:1044::73; FE80::60:3E47:AC8:8 and 2001.DB8:2222:1044::72) of two remote nodes are explicitly mapped to DLCI 17 and DLCI 19, respectively. Both DLCI 17 and DLCI 19 are terminated on interface serial 3 of this node; therefore, interface serial 3 of this node is a point-to-multipoint interface.

```
Router# show frame-relay map
Serial3 (up): ipv6 FE80::E0:F727:E400:A dlci 17(0x11,0x410), static,
              broadcast, CISCO, status defined, active
Serial3 (up): ipv6 2001:DB8:2222:1044::72 dlci 19(0x13,0x430), static,
              CISCO, status defined, active
Serial3 (up): ipv6 2001:DB8:2222:1044::73 dlci 17(0x11,0x410), static,
              CISCO, status defined, active
Serial3 (up): ipv6 FE80::60:3E47:AC8:8 dlci 19(0x13,0x430), static,
              broadcast, CISCO, status defined, active
```

Sample Output from the show hosts Command

The state of the name lookup system on the DHCP for IPv6 client can be displayed with the **show hosts** command:

```
Router# show hosts
Default domain is not set
Domain list:verybigcompany.com
Name/address lookup uses domain service
Name servers are 2001:DB8:A:B::1, 2001:DB8:3000:3000::42
Codes:UN - unknown, EX - expired, OK - OK, ?? - revalidate
```

```
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                    Port  Flags      Age Type   Address(es)
sdfasfd                 None  (temp, UN)  0  IPv6
```

Sample Output from the show running-config Command

In the following example, the **show running-config**command is used to verify that IPv6 processing of packets is enabled globally on the router and on applicable interfaces, and that an IPv6 address is configured on applicable interfaces:

```
Router# show running-config
Building configuration...
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ipv6 unicast-routing
!
interface gigabitethernet0/0/0
 no ip route-cache
 no ip mroute-cache
 no keepalive
 media-type 10BaseT
        ipv6 address 2001:DB8:0:1::/64 eui-64
```

In the following example, the **show running-config** command is used to verify that Cisco Express Forwarding and network accounting for Cisco Express Forwarding have been enabled globally on a nondistributed architecture platform, and that Cisco Express Forwarding has been enabled on an IPv6 interface. The following output shows that both that Cisco Express Forwarding and network accounting for Cisco Express Forwarding have been enabled globally on the router, and that Cisco Express Forwarding has also been enabled on Gigabit Ethernet interface 0/0/0:

```
Router# show running-config
Building configuration...
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface gigabitethernet0/0/0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:DB8:C18:1::/64 eui-64
!
```

In the following example, the **show running-config** command is used to verify static hostname-to-address mappings, default domain names, and name servers in the hostname cache, and to verify that the DNS service is enabled:

```
Router# show running-config
Building configuration...
!
ipv6 host cisco-sj 2001:DB8:20:1::12
!
ip domain-name cisco.com
ip domain-lookup
ip name-server 2001:DB8:C01F:768::1
```

Configuration Examples for Implementing IPv6 Addressing and Basic Connectivity

Example: IPv6 Addressing and IPv6 Routing Configuration

In the following example, IPv6 is enabled on the device with both a link-local address and a global address based on the IPv6 prefix 2001:DB8:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the **show ipv6 interface** command is included to show how the interface ID (260:3EFF:FE47:1530) is appended to the link-local prefix FE80::/64 of Gigabit Ethernet interface 0/0/0.

```
ipv6 unicast-routing
interface gigabitethernet 0/0/0
  ipv6 address 2001:DB8:c18:1::/64 eui-64
Device# show ipv6 interface gigabitethernet 0/0/0
Gigabitethernet0/0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
  Global unicast address(es):
    2001:DB8:C18:1:260:3EFF:FE47:1530, subnet is 2001:DB8:C18:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF47:1530
    FF02::9
  MTU is 1500 bytes
  ICMP error messages limited to one every 500 milliseconds
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Example: Dual-Protocol Stacks Configuration

The following example enables the forwarding of IPv6 unicast datagrams globally on the device and configures Gigabit Ethernet interface 0/0/0 with both an IPv4 address and an IPv6 address:

```
ipv6 unicast-routing
interface gigabitethernet0/0/0
  ip address 192.168.99.1 255.255.255.0
 ipv6 address 2001:DB8:c18:1::3/64
```

Example: IPv6 ICMP Rate Limiting Configuration

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
ipv6 icmp error-interval 50 20
```

Example: Cisco Express Forwarding Configuration

In the following example, both Cisco Express Forwarding for IPv6 and network accounting for Cisco Express Forwarding for IPv6 have been enabled globally on a nondistributed architecture device, and Cisco Express Forwarding for IPv6 has been enabled on Gigabit Ethernet interface 0/0/0. The example also shows that the forwarding of IPv6 unicast datagrams has been configured globally on the device with the **ipv6 unicast-routing** command, an IPv6 address has been configured on Gigabit Ethernet interface 0/0/0 with the **ipv6 address** command, and Cisco Express Forwarding for IPv4 has been configured globally on the device with the **ip cef** command.

```
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
interface gigabitethernet0/0/0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:DB8:C18:1::/64 eui-64
```

Example: Hostname-to-Address Mappings Configuration

The following example defines two static hostname-to-address mappings in the hostname cache, establishes a domain list with several alternate domain names to complete unqualified hostnames, specifies host 2001:DB8::250:8bff:fee8:f800 and host 2001:DB8:0:f004::1 as the name servers, and reenables the DNS service:

```
ip domain list domain1-list.com
ip domain list serviceprovider2-name.com
ip domain list college2-name.edu
ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1
ip domain-lookup
```

Example IPv6 Address to Frame Relay PVC Mapping Configuration

Example IPv6 Frame Relay PVC Mapping Configuration (Point-to-Point Interface)

In the following example, three nodes named Router A, Router B, and Router C make up a fully meshed network. Each node is configured with two PVCs, which provide an individual connection to each of the other two nodes. Each PVC is configured on a different point-to-point subinterface, which creates three unique IPv6 networks (2001:DB8:2222:1017:/64, 2001:DB8:2222:1018::/64, and 2001:DB8:2222:1019::/64). Therefore, the mappings between the IPv6 addresses of each node and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses are implicit (no additional mappings are required).

Note Given that each PVC in the following example is configured on a different point-to-point subinterface, the configuration in the following example can also be used in a network that is not fully meshed. Additionally, configuring each PVC on a different point-to-point subinterface can help simplify your routing protocol configuration. However, the configuration in the following example requires more than one IPv6 network, whereas configuring each PVC on point-to-multipoint interfaces requires only one IPv6 network.

Router A Configuration

```
interface Serial 3
 encapsulation frame-relay
!
interface Serial3.17 point-to-point
 description to Router B
 ipv6 address 2001:DB8:2222:1017::46/64
 frame-relay interface-dlci 17
!
interface Serial 3.19 point-to-point
 description to Router C
 ipv6 address 2001:DB8:2222:1019::46/64
 frame-relay interface-dlci 19
```

Router B Configuration

```
interface Serial 5
 encapsulation frame-relay
!
interface Serial5.17 point-to-point
 description to Router A
 ipv6 address 2001:DB8:2222:1017::73/64
 frame-relay interface-dlci 17
!
interface Serial5.18 point-to-point
 description to Router C
 ipv6 address 2001:DB8:2222:1018::73/64
 frame-relay interface-dlci 18
```

Router C Configuration

```
interface Serial 0
 encapsulation frame-relay
!
interface Serial0.18 point-to-point
 description to Router B
 ipv6 address 2001:DB8:2222:1018::72/64
 frame-relay interface-dlci 18
!
interface Serial0.19 point-to-point
 description to Router A
 ipv6 address 2001:DB8:2222:1019::72/64
 frame-relay interface-dlci 19
```

Example IPv6 Frame Relay PVC Mapping Configuration (Point-to-Multipoint Interface)

In the following example, the same three nodes (Router A, Router B, and Router C) from the previous example make up a fully meshed network and each node is configured with two PVCs (which provide an individual connection to each of the other two nodes). However, the two PVCs on each node in the following example are configured on a single interface (serial 3, serial 5, and serial 10, respectively), which makes each interface a point-to-multipoint interface. Therefore, explicit mappings are required between the link-local and global IPv6 addresses of each interface on all three nodes and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses.

Router A Configuration

```
interface Serial 3
 encapsulation frame-relay
 ipv6 address 2001:DB8:2222:1044::46/64
 frame-relay map ipv6 FE80::E0:F727:E400:A 17 broadcast
 frame-relay map ipv6 FE80::60:3E47:AC8:8 19 broadcast
 frame-relay map ipv6 2001:DB8:2222:1044::72 19
 frame-relay map ipv6 2001:DB8:2222:1044::73 17
```

Router B Configuration

```
interface Serial 5
 encapsulation frame-relay
 ipv6 address 2001:DB8:2222:1044::73/64
 frame-relay map ipv6 FE80::60:3E59:DA78:C 17 broadcast
 frame-relay map ipv6 FE80::60:3E47:AC8:8 18 broadcast
 frame-relay map ipv6 2001:DB8:2222:1044::46 17
 frame-relay map ipv6 2001:DB8:2222:1044::72 18
```

Router C Configuration

```
interface Serial 10
 encapsulation frame-relay
 ipv6 address 2001:DB8:2222:1044::72/64
 frame-relay map ipv6 FE80::60:3E59:DA78:C 19 broadcast
 frame-relay map ipv6 FE80::E0:F727:E400:A 18 broadcast
 frame-relay map ipv6 2001:DB8:2222:1044::46 19
 frame-relay map ipv6 2001:DB8:2222:1044::73 18
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 supported feature list	Start Here: Cisco IOS XE Software Release Specifics for IPv6 Features
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*

Related Topic	Document Title
IPv6 DHCP description and configuration	Implementing DHCP for IPv6
IPv4 addressing configuration tasks	Configuring IPv4 Addresses
IPv4 services configuration tasks	Configuring IP Services
IPv4 addressing commands	*Cisco IOS IP Addressing Services Command Reference*
IPv4 IP services commands	*Cisco IOS IP Application Services Command Reference*
Stateful Switchover	Configuring Stateful Switchover
In Service Software Upgrade	Cisco IOS XE In Service Software Upgrade Process
Switching commands	*Cisco IOS IP Switching Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1981	*Path MTU Discovery for IP version 6*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2461	*Neighbor Discovery for IP Version 6 (IPv6)*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2463	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFCs	Title
RFC 2467	*Transmission of IPv6 Packets over FDDI Networks*
RFC 2472	*IP Version 6 over PPP*
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Networks Specification*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*
RFC 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3596	*DNS Extensions to Support IP version 6*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing IPv6 Addressing and Basic Connectivity

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

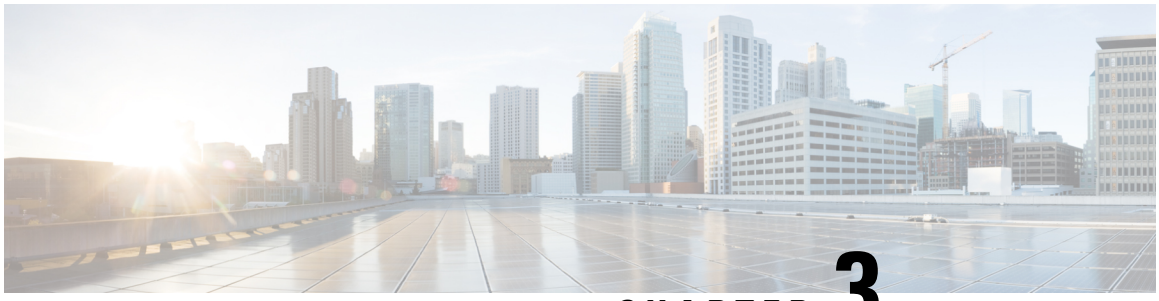
Table 5: Feature Information for Implementing IPv6 Addressing and Basic Connectivity

Feature Name	Releases	Feature Information
IPv6--Base Protocols High Availability	Cisco IOS XE Release 2.1	IPv6 Neighbor Discovery supports SSO.

Feature Name	Releases	Feature Information
IPv6--ICMPv6	Cisco IOS XE Release 2.1	ICMP for IPv6 generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 Neighbor Discovery process, path MTU discovery, and the MLD protocol for IPv6.
IPv6--ICMPv6 Redirect	Cisco IOS XE Release 2.1	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.
IPv6--ICMP Rate Limiting	Cisco IOS XE Release 2.1	The IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network.
IPv6--IPv6 Default Router Preferences	Cisco IOS XE Release 2.1	The DRP extension provides a coarse preference metric (low, medium, or high) for default routers.
IPv6--IPv6 MTU Path Discovery	Cisco IOS XE Release 2.1	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.
IPv6--IPv6 Neighbor Discovery	Cisco IOS XE Release 2.1	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 track neighboring routers.
IPv6--IPv6 Neighbor Discovery Duplicate Address Detection	Cisco IOS XE Release 2.1	IPv6 Neighbor Discovery 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).
IPv6--IPv6 Stateless Autoconfiguration	Cisco IOS XE Release 2.1	The IPv6 stateless autoconfiguration feature can be used to manage link, subnet, and site addressing changes.
IPv6--Per-Interface Neighbor Discovery Cache Limit	Cisco IOS XE Release 2.6	The per-interface Neighbor Discovery cache limit feature provides the ability to limit the number of Neighbor Discovery cache entries on a per interface basis. The following sections provide information about this feature: The following commands were introduced or modified for this feature: **ipv6 nd cache interface-limit (global)** , **ipv6 nd cache interface-limit (interface)**, **show ipv6 neighbors**.

Feature Name	Releases	Feature Information
IPv6--IPv6 Static Cache Entry for Neighbor Discovery	Cisco IOS XE Release 2.1	The IPv6 static cache entry for Neighbor Discovery feature allows static entries to be made in the IPv6 neighbor cache.
IPv6 Data Link--Cisco High-Level Data Link Control (HDLC)	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. HDLC is a type of data link supported for IPv6.
IPv6 Data Link--Fast Ethernet	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. Fast Ethernet data links supported for IPv6.
IPv6 Data Link--FDDI	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. FDDI is a type of data link supported for IPv6.
IPv6 Data Link--Frame Relay PVC	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. Frame relay PVC is a type of data link supported for IPv6.
IPv6 Data Link--PPP Service over Packet over SONET, ISDN, and Serial (Synchronous and Asynchronous) Interfaces	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. PPP service over Packet over SONET, ISDN, and serial interfaces is a type of data link supported for IPv6.
IPv6 Data Link--VLANs Using IEEE 802.1Q Encapsulation	Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. VLANs using IEEE 802.1Q encapsulation is a type of data link supported for IPv6.
IPv6 Services--AAAA DNS Lookups over an IPv4 Transport	Cisco IOS XE Release 2.1	IPv6 basic connectivity can be enhanced by configuring support for AAAA record types in the DNS name-to-address and address-to-name lookup processes.
IPv6 Services--Cisco Discovery Protocol--IPv6 Address Family Support for Neighbor Information	Cisco IOS XE Release 2.1	The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices.
IPv6 Services--DNS Lookups over an IPv6 Transport	Cisco IOS XE Release 2.1	IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes.
IPv6 Switching--Cisco Express Forwarding and Distributed Cisco Express Forwarding Support	Cisco IOS XE Release 2.1	Cisco Express Forwarding for IPv6 is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets. Distributed Cisco Express Forwarding for IPv6 performs the same functions as CEFv6 but for distributed architecture platforms.

Feature Name	Releases	Feature Information
SSO/ISSU Support for per-User IPv6 ACL for PPP Sessions	Cisco IOS XE 3.2.1S	Reproducing IPv6 ACLs on the active RP to the standby RP provides a consistent SSO and ISSU experience for active sessions.
Unicast Reverse Path Forwarding for IPv6	Cisco IOS XE Release 2.1	The Unicast RPF feature mitigates problems caused by malformed or forged (spoofed) IPv6 source addresses that pass through an IPv6 router.



CHAPTER 3

Implementing ADSL for IPv6

This module describes the implementation of prefix pools, the authorization, authentication, and accounting (AAA) server, and per-user Remote Access Dial-In User Service (RADIUS) attributes in IPv6. It also describes the deployment of IPv6 in Digital Subscriber Line (DSL) and dial-access environments. Asymmetric Digital Subscriber Line (ADSL) provides the extensions that make large-scale access possible for IPv6 environments, including IPv6 RADIUS attributes, stateless address configuration on Point-to-Point Protocol (PPP) links, per-user static routes, and access control lists (ACLs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Toolland the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Implementing ADSL for IPv6

ADSL deployment is available for interfaces with PPP encapsulation enabled, including PPP over ATM (PPPoA), PPP over Ethernet (PPPoE, PPPoEoVLAN, PPPoEoQinQ) and PPPoEoA.

Information About Implementing ADSL for IPv6

Address Assignment for IPv6

A Cisco router configured with IPv6 will advertise its IPv6 prefixes on one or more interfaces, allowing IPv6 clients to automatically configure their addresses. In IPv6, address assignment is performed at the network layer, in contrast to IPv4 where a number of functions are handled in the PPP layer. The only function handled in IPv6 Control Protocol is the negotiation of a unique interface identifier. Everything else, including DNS server discovery, is done within the IPv6 protocol itself.

In IPv6, ISPs assign long-lived prefixes to users, which has some impact on the routing system. In typical IPv4 environments, each network access server (NAS) has a pool of 24-bit addresses and users get addresses from this pool when dialing in. If a user dials another POP or is connected to another NAS at the same POP, a different IPv4 address is assigned.

Addresses for IPv6 are assigned by the following methods.

Stateless Address Autoconfiguration

Assigning addresses using the stateless address autoconfiguration method can be used only to assign 64-bit prefixes. Each user is assigned a 64-bit prefix, which is advertised to the user in a router advertisement (RA). All addresses are automatically configured based on the assigned prefix.

A typical scenario is to assign a separate 64-bit prefix per user; however, users can also be assigned a prefix from a shared pool of addresses. Using the shared pool limits addresses to only one address per user.

This method works best for the cases where the customer provider edge (CPE) router is a single PC or is limited to only one subnet. If the user has multiple subnets, Layer 2 (L2) bridging, multilink subnets or proxy RA can be used. The prefix advertised in the RA can come from an authorization, authentication, and accounting (AAA) server, which also provides the prefix attribute, can be manually configured, or can be allocated from a prefix pool.

The Framed-Interface-Id AAA attribute influences the choice of interface identifier for peers and, in combination with the prefix, the complete IPv6 address can be determined.

Prefix Delegation

An IPv6 prefix delegating device selects IPv6 prefixes to be assigned to a requesting device upon receiving a request from the client. The delegating device might select prefixes for a requesting device in the following ways:

- Dynamic assignment from a pool of available prefixes.
- Dynamic assignment from a pool name obtained from the RADIUS server.
- Assignment of prefix obtained from the RADIUS sever.

Contrary to IPv4 address assignment, an IPv6 user will be assigned a prefix, not a single address. Typically the Internet service provider (ISP) assigns a 64- or 48-bit prefix.

Accounting Start and Stop Messages

PPP calls a registry to allow DHCPv6 to append the delegated prefix information to accounting start and stop messages. When accounting is configured for a DHCPv6 pool, accounting interim packets are sent to broadband sessions after binding is provided from the pool.

Forced Release of a Binding

The DHCPv6 server maintains an automatic binding table in memory to track the assignment of some configuration parameters, such as prefixes between the server and its clients. The automatic bindings can be stored permanently in the database agent, which can be, for example, a remote TFTP server or local NVRAM file system.

DHCPv6 invokes a routine when the virtual interface used by PPP terminates. This routine automatically releases any delegated prefix bindings associated with the PPP virtual interface that is being terminated.

When a PPP virtual interface terminates, the routine runs through the full table of DHCPv6 bindings checking for the matching interface. Because PPP uses a virtual interface, this subroutine clears any related lease information when the PPP connection terminates.

Note In IPv6 broadband deployment using DHCPv6, you must enable release of prefix bindings associated with a PPP virtual interface using the **ipv6 dhcp binding track ppp** command. This ensures that DHCPv6 bindings are tracked together with PPP sessions, and in the event of DHCP REBIND failure, the client initiates DHCPv6 negotiation again.

DHCP SIP Server Options

Two DHCP for IPv6 Session Initiation Protocol (SIP) server options describe a local outbound SIP proxy: one carries a list of domain names, the other a list of IPv6 addresses. These two options can be configured in a DHCPv6 configuration pool.

AAA over IPv6

Vendor-specific attributes (VSAs) are used to support Authentication, Authorization and Accounting(AAA) over IPv6. Cisco VSAs are inacl, outacl, prefix, and route.

You can configure prefix pools and pool names by using the AAA protocol. Customers can deploy an IPv6 RADIUS server or a TACACS+ server to communicate with Cisco devices.

AAA Support for IPv6 RADIUS Attributes

The following RADIUS attributes, as described in RFC 3162, are supported for IPv6:

- Framed-Interface-Id
- Framed-IPv6-Pool
- Framed-IPv6-Prefix
- Framed-IPv6-Route
- Login-IPv6-Host

The following RADIUS attributes are also supported for IPv6:

- Delegated-IPv6-Prefix (RFC 4818)
- Delegated-IPv6-Prefix-Pool
- DNS-Server-IPv6-Address
- IPv6 ACL
- IPv6_DNS_Servers
- IPv6 Pool
- IPv6 Prefix#
- IPv6 Route

The attributes listed above can be configured on a RADIUS server and downloaded to access servers, where they can be applied to access connections.

Prerequisites for Using AAA Attributes for IPv6

AAA attributes for IPv6 are compliant with RFC 3162 and require a RADIUS server capable of supporting RFC 3162.

RADIUS Per-User Attributes for Virtual Access in IPv6 Environments

The following IPv6 RADIUS attributes are supported for virtual access and can be used as attribute-value (AV) pairs:

- Delegated-IPv6-Prefix
- Delegated-IPv6-Prefix-Pool
- DNS-Server-IPv6-Address
- Framed-Interface-Id
- Framed-IPv6-Pool
- Framed-IPv6-Prefix
- Framed-IPv6-Route
- IPv6 ACL
- IPv6_DNS_Servers
- IPv6 Pool
- IPv6 Prefix#
- IPv6 Route
- Login-IPv6-Host

Delegated-IPv6-Prefix

The Delegated-IPv6-Prefix attribute indicates an IPv6 prefix to be delegated to a user for use in a network. This attribute is used during DHCP prefix delegation between a RADIUS server and a delegating device. A Network Access Server (NAS) that hosts a DHCP Version 6 (DHCPv6) server can act as a delegating device.

The following example shows how to use the Delegated-IPv6-Prefix attribute:

```
ipv6:delegated-prefix=2001:DB8::/64
```

Note The Cisco VSA format is not supported for this attribute. If you try to add this attribute in the Cisco VSA format into a user profile, the RADIUS server response fails. Use only the IETF attribute format for this attribute.

Delegated-IPv6-Prefix-Pool

The Delegated-IPv6-Prefix-Pool attribute indicates the name of a prefix pool from which a prefix is selected and delegated to a device.

Prefix delegation is a DHCPv6 option for delegating IPv6 prefixes. Prefix delegation involves a delegating device that selects a prefix and assigns it on a temporary basis to a requesting device. A delegating device uses many strategies to choose a prefix. One method is to choose a prefix from a prefix pool with a name that is defined locally on a device.

The Delegated-IPv6-Prefix-Pool attribute indicates the name of an assigned prefix pool. A RADIUS server uses this attribute to communicate the name of a prefix pool to a NAS hosting a DHCPv6 server and acting as a delegating device.

You may use DHCPv6 prefix delegation along with ICMPv6 stateless address autoconfiguration (SLAAC) on a network. In this case, both the Delegated-IPv6-Prefix-Pool attribute and the Framed-IPv6-Pool attribute may be included within the same packet. To avoid ambiguity, the Delegated-IPv6-Prefix-Pool attribute should be restricted to the authorization and accounting of prefix pools used in DHCPv6 delegation, and the Framed-IPv6-Pool attribute should be used for the authorization and accounting of prefix pools used in SLAAC.

The following example shows how an address prefix is selected from a pool named pool1. The prefix pool pool1 is downloaded to a delegating device from a RADIUS server by using the Delegated-IPv6-Prefix-Pool attribute. The device then selects the address prefix 2001:DB8::/64 from this prefix pool.

```
Cisco:Cisco-AVpair = "ipv6:delegated-ipv6-pool = pool1"
!
ipv6 dhcp pool pool1
address prefix 2001:DB8::/64
!
```

DNS-Server-IPv6-Address

The DNS-Server-IPv6-Address attribute indicates the IPv6 address of a Domain Name System (DNS) server. A DHCPv6 server can configure a host with the IPv6 address of a DNS server. The IPv6 address of the DNS server can also be conveyed to the host using router advertisement messages from ICMPv6 devices.

A NAS may host a DHCPv6 server to handle DHCPv6 requests from hosts. The NAS may also act as a device that provides router advertisement messages. Therefore, this attribute is used to provide the NAS with the IPv6 address of the DNS server.

If a NAS has to announce more than one recursive DNS server to a host, this attribute can be included multiple times in Access-Accept packets sent from the NAS to the host.

The following example shows how you can define the IPv6 address of a DNS server by using the DNS-Server-IPv6-Address attribute:

```
Cisco:Cisco-AVpair = "ipv6:ipv6-dns-servers-addr=2001:DB8::"
```

Framed-Interface-Id

The Framed-Interface-Id attribute indicates an IPv6 interface identifier to be configured for a user.

This attribute is used during IPv6 Control Protocol (IPv6CP) negotiations of the Interface-Identifier option. If negotiations are successful, the NAS uses this attribute to communicate a preferred IPv6 interface identifier to the RADIUS server by using Access-Request packets. This attribute may also be used in Access-Accept packets.

Framed-IPv6-Pool

The Framed-IPv6-Pool attribute indicates the name of a pool that is used to assign an IPv6 prefix to a user. This pool should be either defined locally on a device or defined on a RADIUS server from where pools can be downloaded.

Framed-IPv6-Prefix

The Framed-IPv6-Prefix attribute indicates an IPv6 prefix (and a corresponding route) to be configured for a user. So this attribute performs the same function as a Cisco VSA and is used for virtual access only. A NAS uses this attribute to communicate a preferred IPv6 prefix to a RADIUS server by using Access-Request packets. This attribute may also be used in Access-Accept packets and can appear multiple times in these packets. The NAS creates a corresponding route for the prefix.

This attribute is used by a user to specify which prefixes to advertise in router advertisement messages of the Neighbor Discovery Protocol.

This attribute can also be used for DHCPv6 prefix delegation, and a separate profile must be created for a user on the RADIUS server. The username associated with this separate profile has the suffix “-dhcpv6”.

The Framed-IPv6-Prefix attribute is treated differently in this separate profile and the regular profile of a user. If a NAS needs to send a prefix through router advertisement messages, the prefix is placed in the Framed-IPv6-Prefix attribute of the regular profile of the user. If a NAS needs to delegate a prefix to the network of a remote user, the prefix is placed in the Framed-IPv6-Prefix attribute of the separate profile of the user.

Note The RADIUS IETF attribute format and the Cisco VSA format are supported for this attribute.

Framed-IPv6-Route

The Framed-IPv6-Route attribute indicates the routing information to be configured for a user on a NAS. This attribute performs the same function as a Cisco VSA. The value of the attribute is a string and is specified by using the **ipv6 route** command.

IPv6 ACL

The IPv6 ACL attribute is used to specify a complete IPv6 access list. The unique name of an access list is generated automatically. An access list is removed when the respective user logs out. The previous access list on the interface is then reapplied.

The inacl and outacl attributes enable you to specify an existing access list configured on a device. The following example shows how to define an access list identified with number 1:

```
cisco-avpair = "ipv6:inacl#1=permit 2001:DB8:cc00:1::/48",
cisco-avpair = "ipv6:outacl#1=deny 2001:DB8::/10",
```

IPv6_DNS_Servers

The IPv6_DNS_Servers attribute is used to send up to two DNS server addresses to the DHCPv6 server. The DNS server addresses are saved in the interface DHCPv6 subblock and override other configurations in the DHCPv6 pool. This attribute is also included in attributes returned for AAA start and stop notifications.

IPv6 Pool

The IPv6 Pool attribute extends the IPv4 address pool attribute to support the IPv6 protocol for RADIUS authentication. This attribute specifies the name of a local pool on a NAS from which a prefix is chosen and used whenever PPP is configured and the protocol is specified as IPv6. The address pool works with local pooling and specifies the name of a local pool that is preconfigured on the NAS.

IPv6 Prefix#

The IPv6 Prefix# attribute indicates which prefixes to advertise in router advertisement messages of the Neighbor Discovery Protocol. When this attribute is used, a corresponding route (marked as a per-user static route) is installed in the routing information base (RIB) tables for a given prefix.

The following example shows how to specify which prefixes to advertise:

```
cisco-avpair = "ipv6:prefix#1=2001:DB8::/64",
cisco-avpair = "ipv6:prefix#2=2001:DB8::/64",
```

IPv6 Route

The IPv6 Route attribute is used to specify a static route for a user. A static route is appropriate when Cisco software cannot dynamically build a route to the destination. See the **ipv6 route** command for more information about building static routes.

The following example shows how to use the IPv6 Route attribute to define a static route:

```
cisco-avpair = "ipv6:route#1=2001:DB8:cc00:1::/48",
cisco-avpair = "ipv6:route#2=2001:DB8:cc00:2::/48",
```

Login-IPv6-Host

The Login-IPv6-Host attribute indicates IPv6 addresses of hosts with which to connect a user when the Login-Service attribute is included. A NAS uses the Login-IPv6-Host attribute in Access-Request packets to communicate to a RADIUS server that it prefers to use certain hosts.

PPP IPv6 Accounting Delay Enhancements

This feature enhances accounting records for dual-stack networks. It ensures that a unique IPv6 address is assigned to PPP IPv6 and IPv4 sessions for IP addresses that are received from RADIUS.

When this feature is enabled, it automatically creates a database to hold new incoming access-accept responses from RADIUS. The access-accept responses in this database are then checked for duplicates of a specific set of attributes. If the attributes are already present in the database, then the RADIUS server has already offered them to an existing session; therefore, the new session is immediately removed and a stop-record message sent. If none of the specific set of attributes are in the database, they are immediately added to the database, and the session proceeds normally. When the session is removed, the entries in the database are also removed.

The following RADIUS attributes are tracked in the database and checked at access-accept time:

- Framed-IPv6-Prefix
- Delegated-IPv6-Prefix

The attributes are available as standard RFC-defined binary format, or as Cisco VSAs. (The Delegated-IPv6-Prefix attribute currently does not have a VSA definition in AAA.)

TACACS+ Over an IPv6 Transport

An IPv6 server can be configured to use TACACS+. Both IPv6 and IPv4 servers can be configured to use TACACS+ using a name instead of an IPv4 or IPv6 address.

IPv6 Prefix Pools

The function of prefix pools in IPv6 is similar to that of address pools in IPv4. The main difference is that IPv6 assigns prefixes rather than single addresses.

As in IPv4, a pool or a pool definition in IPv6 can be configured locally or it can be retrieved from an AAA server. Overlapping membership between pools is not permitted.

Once a pool is configured, it cannot be changed. If you change the configuration, the pool will be removed and re-created. All prefixes previously allocated will be freed.

Prefix pools can be defined so that each user is allocated a 64-bit prefix or so that a single prefix is shared among several users. In a shared prefix pool, each user may receive only one address from the pool.

Broadband IPv6 Counter Support at LNS

This feature provides support for broadband PPP IPv6 sessions at the layer 2 tunneling protocol (L2TP) network server (LNS). The sessions are forwarded by L2TP access concentrator (LAC) using layer 2 tunneling protocol L2TP over IPv6.

This feature is enabled automatically when the user configures LNS and enables IPv6.

How to Configure ADSL in IPv6

Configuring the NAS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **aaa new-model**
5. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
6. **aaa authorization configuration default** {**radius** | **tacacs+**
7. **show ipv6 route** [*ipv6-address* | *ipv6-prefix / prefix-length* | *protocol* | *interface-type interface-number*
8. **virtual-profile virtual-template** *number*
9. **interface serial** *controller-number* **:** *timeslot*
10. **encapsulation** *encapsulation-type*
11. **exit**
12. **dialer-group** *group-number*
13. **ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**]
14. **interface virtual-template** *number*
15. **ipv6 enable**
16. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | *access-group*}
17. **radius-server host** {*hostname* | *ip-address*} [**test username** *user-name*] [**auth-port** *port-number*] [**ignore-auth-port**] [**acct-port** *port-number*] [**ignore-acct-port**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] [**idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *name* **Example:**	Specifies the hostname for the network server.

	Command or Action	Purpose
	`Router(config)# hostname cust1-53a`	
Step 4	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA server.
Step 5	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp default if-needed group radius`	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.
Step 6	**aaa authorization configuration default** {**radius** | **tacacs+** **Example:** `Router(config)# aaa authorization configuration default radius`	Downloads configuration information from the AAA server.
Step 7	**show ipv6 route** [*ipv6-address* | *ipv6-prefix* / *prefix-length* | *protocol* | *interface-type interface-number* **Example:** `Router(config)# show ipv6 route`	Shows the routes installed by the previous commands.
Step 8	**virtual-profile virtual-template** *number* **Example:** `Router(config)# virtual-profile virtual-template 1`	Enables virtual profiles by virtual interface template.
Step 9	**interface serial** *controller-number* **:** *timeslot* **Example:** `Router(config)# interface serial 0:15`	Specifies a serial interface created on a channelized E1 or channelized T1 controller (for ISDN PRI, channel-associated signaling, or robbed-bit signaling). This command also puts the router into interface configuration mode.
Step 10	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 12	**dialer-group** *group-number* **Example:** `Router(config)# dialer-group 1`	Controls access by configuring an interface to belong to a specific dialing group.
Step 13	**ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**] **Example:** `Router(config)# ppp authentication chap`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 14	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
Step 15	**ipv6 enable** **Example:** `Router(config)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 16	**dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | *access-group*} **Example:** `Router(config)# dialer-list 1 protocol ipv6 permit`	Defines a dial-on-demand routing (DDR) dialer list for dialing by protocol or by a combination of a protocol and a previously defined access list.
Step 17	**radius-server host** {*hostname* | *ip-address*} [**test username** *user-name*] [**auth-port** *port-number*] [**ignore-auth-port**] [**acct-port** *port-number*] [**ignore-acct-port**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] [**idle-time** *seconds* **Example:** `Router(config)# radius-server host 172.17.250.8 auth-port 1812 acct-port 1813 key testing123`	Specifies a RADIUS server host.

Enabling the Sending of Accounting Start and Stop Messages

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **accounting** *mlist*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCP for IPv6 configuration information pool and enters DHCP for IPv6 pool configuration mode.
Step 4	**accounting** *mlist* **Example:** `Device(config-dhcp)# accounting list1`	Enables accounting start and stop messages to be sent.

Removing Delegated Prefix Bindings

Perform this task to release any delegated prefix bindings associated with the PPP virtual interface that is being terminated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp bindings track ppp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface VirtualAccess2.2`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 dhcp bindings track ppp** **Example:** `Device(config-if)# ipv6 dhcp bindings track ppp`	Releases any delegated prefix leases associated with the PPP virtual interface that is being terminated.

Configuring DHCPv6 AAA Options

Perform the following task to configure the option of acquiring prefixes from the AAA server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *pool-name*
4. **prefix-delegation aaa** [**method-list** *method-list*] [*lifetime*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *pool-name* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters IPv6 DHCP pool configuration mode.
Step 4	**prefix-delegation aaa** [**method-list** *method-list*] [*lifetime*] **Example:** `Device(config-dhcpv6)# prefix-delegation aaa method-list list1`	Specifies that prefixes are to be acquired from AAA servers.
Step 5	**end** **Example:** `Device(config-dhcpv6)# end`	Exits IPv6 DHCP pool configuration mode and returns to privileged EXEC mode.

Configuring PPP IPv6 Accounting Delay Enhancements

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ppp unique address access-accept**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ppp unique address access-accept** **Example:** `Router(config)# ppp unique address access-accept`	Tracks duplicate addresses received from RADIUS and creates a standalone database.

Configuring TACACS+ over IPv6

Configuring the TACACS+ Server over IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **tacacs server** *name*
4. **address ipv6** *ipv6-address*
5. **key** [**0** | **7**] *key-string*
6. **port** [*number*
7. **send-nat-address**
8. **single-connection**
9. **timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**tacacs server** *name* **Example:** `Device(config)# tacacs server server1`	Configures the TACACS+ server for IPv6 and enters TACACS+ server configuration mode.
Step 4	**address ipv6** *ipv6-address* **Example:** `Device(config-server-tacacs)# address ipv6 2001:DB8:3333:4::5`	Configures the IPv6 address of the TACACS+ server.
Step 5	**key** [**0** | **7**] *key-string* **Example:** `Device(config-server-tacacs)# key 0 key1`	Configures the per-server encryption key on the TACACS+ server.
Step 6	**port** [*number* **Example:** `Device(config-server-tacacs)# port 12`	Specifies the TCP port to be used for TACACS+ connections.
Step 7	**send-nat-address** **Example:** `Device(config-server-tacacs)# send-nat-address`	Sends a client's post-NAT address to the TACACS+ server.
Step 8	**single-connection** **Example:** `Device(config-server-tacacs)# single-connection`	Enables all TACACS packets to be sent to the same server using a single TCP connection.
Step 9	**timeout** *seconds* **Example:** `Device(config-server-tacacs)# timeout 10`	Configures the time to wait for a reply from the specified TACACS server.

Specifying the Source Address in TACACS+ Packets

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 tacacs source-interface** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 tacacs source-interface** *type number* **Example:** `Router(config)# ipv6 tacacs source-interface GigabitEthernet 0/0/0`	Specifies an interface to use for the source address in TACACS+ packets.

Configuring TACACS+ Server Group Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server tacacs+** *group-name*
4. **server name** *server-name*
5. **server-private** {*ip-address* | *name* | *ipv6-address*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server tacacs+** *group-name* **Example:** `Device(config)# aaa group server tacacs+ group1`	Groups different TACACS+ server hosts into distinct lists and distinct methods.
Step 4	**server name** *server-name* **Example:** `Device(config-sg-tacacs+)# server name server1`	Specifies an IPv6 TACACS+ server.
Step 5	**server-private** {*ip-address* | *name* | *ipv6-address*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Device(config-sg-tacacs+)# server-private 2001:DB8:3333:4::5 port 19 key key1`	Configures the IPv6 address of the private TACACS+ server for the group server.

Verifying Broadband IPv6 Counter Support at the LNS

This feature is enabled automatically when the user configures LNS and enables IPv6. To verify information about this feature, you can use any or all of the following optional commands as needed.

SUMMARY STEPS

1. **enable**
2. **show l2tp session** [**all** | **packets** [**ipv6**] | **sequence** | **state** | [**brief** | **circuit** | **interworking**] [**hostname**]] [**ip-addr** *ip-addr*[**vcid** *vcid*] | **tunnel**{**id** *local-tunnel-id local-session-id*| **remote-name** *remote-tunnel-name local-tunnel-name*}| **username** *username* | **vcid** *vcid*]
3. **show l2tp tunnel** [**all** | **packets** [**ipv6**] | **state** | **summary** | **transport**] [**id** *local-tunnel-id* | **local-name** *local-tunnel-name remote-tunnel-name*| **remote-name** *remote-tunnel-name local-tunnel-name*]
4. **show l2tun session** [**l2tp** | **pptp**] [**all** [*filter*] | **brief** [*filter*] [**hostname**] | **circuit** [*filter*] [**hostname**] | **interworking** [*filter*] [**hostname**] | **packets ipv6**] [*filter*] | **sequence** [*filter*] | **state** [*filter*]]
5. **show vpdn session** [**l2f** | **l2tp** | **pptp**] [**all** | **packets** [**ipv6**] | **sequence** | **state** [*filter*]]
6. **show vpdn tunnel** [**l2f** | **l2tp** | **pptp**] [**all** [*filter*] | **packets ipv6**] [*filter*] | **state** [*filter*] | **summary** [*filter*] | **transport**[*filter*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**show l2tp session** [**all** | **packets** [**ipv6**] | **sequence** | **state** | [**brief** | **circuit** | **interworking**] [**hostname**]] [**ip-addr** *ip-addr*[**vcid** *vcid*] | **tunnel**{**id** *local-tunnel-id local-session-id*| **remote-name** *remote-tunnel-name local-tunnel-name*}| **username** *username* | **vcid** *vcid*] **Example:** `Router# show l2tp session packets ipv6`	Displays information about L2TP sessions.
Step 3	**show l2tp tunnel** [**all** | **packets** [**ipv6**] | **state** | **summary** | **transport**] [**id** *local-tunnel-id* | **local-name** *local-tunnel-name remote-tunnel-name*| **remote-name** *remote-tunnel-name local-tunnel-name*] **Example:** `Router# show l2tp tunnel packets ipv6`	Displays details about L2TP tunnels.
Step 4	**show l2tun session** [**l2tp** | **pptp**] [**all** [*filter*] | **brief** [*filter*] [**hostname**] | **circuit** [*filter*] [**hostname**] | **interworking** [*filter*] [**hostname**] | **packets ipv6**] [*filter*] | **sequence** [*filter*] | **state** [*filter*]] **Example:** `Router# show l2tun session packets ipv6`	Displays the current state of Layer 2 sessions and protocol information about L2TP control channels.
Step 5	**show vpdn session** [**l2f** | **l2tp** | **pptp**] [**all** | **packets** [**ipv6**] | **sequence** | **state** [*filter*]] **Example:** `Router# show vpdn session packets ipv6`	Displays session information about active Layer 2 sessions for a virtual private dialup network (VPDN).
Step 6	**show vpdn tunnel** [**l2f** | **l2tp** | **pptp**] [**all** [*filter*] | **packets ipv6**] [*filter*] | **state** [*filter*] | **summary** [*filter*] | **transport**[*filter*]] **Example:** `Router# show vpdn tunnel packets ipv6`	Displays information about active Layer 2 tunnels for a VPDN.

Configuration Examples for Implementing ADSL for IPv6

Example NAS Configuration

This configuration for the ISP NAS shows the configuration that supports access from the remote CE router.

```
hostname hostname1
  aaa new-model
  aaa authentication ppp default if-needed group radius
  aaa authorization network default

  aaa accounting network default start-stop group radius

  aaa accounting send counters ipv6

  interface virtual-template 1

  ip unnumbered loopback interface1

  ipv6 address autoconfig

  no ipv6 nd ra suppress
  ppp authentication chap

  ppp accounting list1

  no snmp trap link-status

  no logging event link-status

  exit

  aaa group service radius group1

  server-private 10.1.1.1 timeout 5 retransmit 3 key xyz

  radius-server host 192.0.2.176 test username test1 auth-port 1645 acct-port 1646

  radius-server vsa send accounting

  radius-server vsa send authentication
```

Example RADIUS Configuration

This RADIUS configuration shows the definition of AV pairs to establish the static routes.

```
  campus1 Auth-Type = Local, Password = "mypassword"
```

```
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ipv6:inacl#1=permit dead::/64 any",
cisco-avpair = "ipv6:route=library::/64",
cisco-avpair = "ipv6:route=cafe::/64",
cisco-avpair = "ipv6:prefix=library::/64 0 0 onlink autoconfig",
cisco-avpair = "ipv6:prefix=cafe::/64 0 0 onlink autoconfig",
cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
```

Examples: Verifying Broadband IPv6 Counter Support at the LNS

Example: show l2tp session Command

The **show l2tp session** command used with the **packets and ipv6**keywords displays information about IPv6 packets and byte counts in an L2TP session.

```
Router# show l2tp session packets ipv6


L2TP Session Information Total tunnels 1 sessions 1


LocID      RemID      TunID      Pkts-In    Pkts-Out   Bytes-In   Bytes-Out

16791      53352      27723      30301740   30301742   20159754280 20523375360
```

Example: show l2tp tunnel Command

The **show l2tp tunnel** command used with the **packets** and **ipv6** keywords displays information about IPv6 packet statistics and byte counts in L2TP tunnels.

```
Router# show l2tp tunnel packets ipv6
L2TP Tunnel Information Total tunnels 1 sessions 1
LocTunID   Pkts-In    Pkts-Out   Bytes-In   Bytes-Out
27723      63060379   63060383   39400320490 40157045438
```

Example: show l2tun session Command

The **show l2tun session** command used with the **packets**and **ipv6**keywords displays information about IPv6 packet statistics and byte counts in an L2TUN session.

```
Router# show l2tun session packets ipv6
L2TP Session Information Total tunnels 1 sessions 1
LocID      RemID      TunID      Pkts-In    Pkts-Out   Bytes-In   Bytes-Out
16791      53352      27723      31120707   31120708   21285014938 21658462236
```

Example: show vpdn session Command

The **show vpdn session** command used with the **l2tp**, **packets**, and **ipv6**keywords displays session information about IPv6 packet statistics and byte counts in an active layer 2 session for a VPDN.

```
Router# show vpdn session l2tp packets ipv6
```

```
L2TP Session Information Total tunnels 1 sessions 1
LocID       RemID       TunID       Pkts-In     Pkts-Out    Bytes-In    Bytes-Out
16791       53352       27723       35215536    35215538    22616342688 23038929320
```

Example: show vpdn tunnel Command

The **show vpdn tunnel**command used with the **l2tp**, **packets**, and **ipv6**keywords displays session information about IPv6 packet statistics and byte counts in an active layer 2 tunnel for a VPDN.

```
Device# show vpdn tunnel l2tp packets ipv6
L2TP Tunnel Information Total tunnels 1 sessions 1
LocTunID    Pkts-In     Pkts-Out    Bytes-In    Bytes-Out
27723       61422447    61422451    37149801922 37886871686
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 supported feature list	" Start Here: Cisco IOS XE Software Release Specifics for IPv6 Features ," *Cisco IOS XE IPv6 Configuration Guide*
IPv6 basic connectivity	" Implementing IPv6 Addressing and Basic Connectivity, " *Cisco IOS XE IPv6 Configuration Guide*
DHCP for IPv6	" Implementing DHCP for IPv6, " *Cisco IOS XE IPv6 Configuration Guide*
IPv6 commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3162	*RADIUS and IPv6*
RFC 3177	*IAB/IESG Recommendations on IPv6 Address*
RFC 3319	*Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing ADSL for IPv6

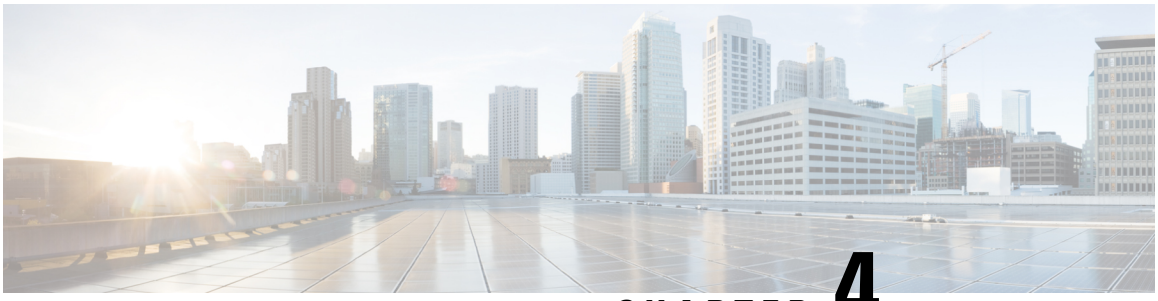
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Implementing ADSL for IPv6

Feature Name	Releases	Feature Information
Enhanced IPv6 Features for ADSL and Dial Deployment	Cisco IOS XE Release 2.5	Several features were enhanced to enable IPv6 to use ADSL and dial deployment.
AAA Support for Cisco VSA IPv6 Attributes	Cisco IOS XE Release 2.5	Vendor-specific attributes (VSAs) were developed to support AAA for IPv6.
IPv6 Access Services: PPPoE	Cisco IOS XE Release 2.5	ADSL and dial deployment is available for interfaces with PPP encapsulation enabled, including PPPoE.
AAA Support for RFC 3162 IPv6 RADIUS Attributes	Cisco IOS XE Release 2.5	The AAA attributes for IPv6 are compliant with RFC 3162 and require a RADIUS server capable of supporting RFC 3162. The following commands were modified by this feature: **ipv6 dhcp pool**, prefix-delegation aaa

Feature Name	Releases	Feature Information
DHCP - DHCPv6 Prefix Delegation RADIUS VSA	Cisco IOS XE Release 2.5	When the user requests a prefix from the prefix delegator, typically the NAS, the prefix is allocated using DHCPv6.
PPP Enhancement for Broadband IPv6	Cisco IOS XE Release 2.5	The following sections provide information about this feature.
AAA Improvements for Broadband IPv6	Cisco IOS XE Release 2.5	
DHCP Enhancements to Support IPv6 Broadband Deployments	Cisco IOS XE Release 2.5	
PPPoA	Cisco IOS XE Release 3.3S	ADSL and dial deployment is available for interfaces with PPP encapsulation enabled, including PPPoA.
SSO - PPPoE IPv6	Cisco IOS XE Release 2.5	This feature is supported in Cisco IOS XE Release 2.5.
Broadband IPv6 Counter Support at LNS	Cisco IOS XE Release 2.6	This feature provides support for broadband PPP IPv6 sessions at the L2TP LNS. The sessions are forwarded by LAC using layer 2 tunneling protocol L2TP over IPv4. The following commands were modified by this feature: **show l2tp session**, **show l2tp tunnel**, **show l2tun session**, **show vpdn session**, **show vpdn tunnel**.
PPP IPv6 Accounting Delay Enhancements	Cisco IOS XE Release 3.2S	This feature enhances accounting records for dual-stack networks. It ensures that a unique IPv6 address is assigned to PPP IPv6 and IPv4 sessions for IP addresses that are received from RADIUS. The following command was introduced by this feature: **debug ppp unique address**, **ppp unique address access-accept**
RADIUS over IPv6	Cisco IOS XE Release 3.2S	RADIUS over IPv6 is supported.
TACACS+ over IPv6	Cisco IOS XE Release 3.2S	TACACS+ over IPv6 is supported. The following commands were introduced or modified by this feature: **aaa group server tacacs+**, **address ipv6 (TACACS+)**, **ipv6 tacacs source-interface**, **key (TACACS+)**, **port (TACACS+)**, **send-nat-address**, **server name (IPv6 TACACS+)**, **server-private (TACACS+)**, **single-connection**, **tacacs server**, **timeout (TACACS+)**.



CHAPTER 4

Implementing Bidirectional Forwarding Detection for IPv6

This document describes how to implement the Bidirectional Forwarding Detection for IPv6 (BFDv6) protocol. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. BFDv6 provides IPv6 support by accommodating IPv6 addresses, and it provides the ability to create BFDv6 sessions.

Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Toland the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Implementing Bidirectional Forwarding Detection for IPv6

IPv6 Cisco Express Forwarding and IPv6 unicast routing must be enabled on all participating routers.

Restrictions for Implementing Bidirectional Forwarding Detection for IPv6

- BFDv6 supports only global IPv6 neighbor addresses if a global IPv6 address is configured on the interface.
- Only asynchronous mode is supported. In asynchronous mode, either BFDv6 peer can initiate a BFDv6 session.

Information About Implementing Bidirectional Forwarding Detection for IPv6

Overview of the BFDv6 Protocol

This section describes the BFDv6 protocol, how it is different from BFD for IPv4, and how it works with BFD for IPv4. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. BFDv6 provides IPv6 support by accommodating IPv6 addresses and provides the ability to create BFDv6 sessions.

BFDv6 Registration

BFD clients register with BFD using a registry application program interface (API). The registry arguments include protocol type and the address and interface description block (IDB) of the route to be monitored. These APIs and arguments are all assumed by BFD to be IPv4.

BFDv6 has registries from which these arguments have been removed, and the protocol and encapsulation are described within a session information structure. These session information structures are defined by BFDv6 for the protocols supported. BFDv6 uses information from the session information structures to determine the correct encapsulation for BFDv6 packets on that session.

BFDv6 Global and Link-Local Addresses

BFDv6 supports both global and link-local IPv6 addresses for neighbor creation. BFDv6 sessions select source addresses to match the neighbor address types (for example, global IPv6 address neighbors must be paired with global IPv6 source addresses and link-local IPv6 address neighbors must be paired with link-local IPv6 source addresses). The table below shows the address pairings that BFDv6 supports.

Table 7: BFDv6 Address Pairings for Neighbor Creation

Source Address	Destination Address	Status
Global	Global	Supported
Global	Link local	Not supported
Link local	Global	Not supported
Link local	Link local	Supported

Because all IPv6-enabled interfaces have a link-local address and BFDv6 selects the source address, link-local address neighbors are always paired with a link-local interface address. The link-local source address with global destination address is not supported by Cisco Express Forwarding. Therefore, a global IPv6 address must be configured on an interface before a session with a global address neighbor may be established in BFDv6. BFDv6 rejects any sessions in which the neighbor address is global and no global address is configured on the interface.

Note The behavior of a unique local address (ULA) in BFDv6 is the same as a global address.

BFD for IPv4 and IPv6 on the Same Interface

BFD supports multiple IPv4 and IPv6 sessions per interface, with no restriction on the protocol of those sessions.

Static Route Support for BFD over IPv6

Using the BFDv6 protocol to reach the static route next hop ensures that an IPv6 static route is inserted only in the IPv6 Routing Information Base (RIB) when the next-hop neighbor is reachable. Using the BFDv6 protocol also can remove the IPv6 static route from the IPv6 RIB when the next hop becomes unreachable.

You can configure IPv6 static BFDv6 neighbors. These neighbors can operate in one of two modes: associated (which is the default) and unassociated. A neighbor can be transitioned between the two modes without interrupting the BFDv6 session associated with the neighbor.

BFDv6 Associated Mode

In Bidirectional Forwarding Detection for IPv6 (BFDv6) associated mode, an IPv6 static route is automatically associated with an IPv6 static BFDv6 neighbor if the static route next hop exactly matches the static BFDv6 neighbor.

An IPv6 static route requests a BFDv6 session for each static BFDv6 neighbor that has one or more associated IPv6 static routes and is configured over an interface on which BFD has been configured. The state of the BFDv6 session will be used to determine whether the associated IPv6 static routes are inserted in the IPv6 RIB. For example, static routes are inserted in the IPv6 RIB only if the BFDv6 neighbor is reachable, and the static route is removed from the IPv6 RIB if the BFDv6 neighbor subsequently becomes unreachable.

BFDv6 associated mode requires you to configure a BFD neighbor and static route on both the device on which the BFD-monitored static route is required and on the neighboring device.

BFDv6 Unassociated Mode

An IPv6 static BFD neighbor may be configured as unassociated. In this mode, the neighbor is not associated with static routes, and the neighbor always requests a BFDv6 session if the interface has been configured for BFDv6.

Unassociated mode is useful in the following situations:

- Bringing up a BFDv6 session in the absence of an IPv6 static route—This case occurs when a static route is on router A, with router B as the next hop. Associated mode requires you to create both a static BFD neighbor and static route on both routers in order to bring up the BFDv6 session from B to A. Specifying the static BFD neighbor in unassociated mode on router B avoids the need to configure an unwanted static route.
- Transition to BFD monitoring of a static route—This case occurs when existing IPv6 static routes are inserted in the IPv6 RIB. Here, you want to enable BFD monitoring for these static routes without any interruption to traffic. If you configure an attached IPv6 static BFD neighbor, then the static routes will immediately be associated with the new static BFD neighbor. However, because a static BFD neighbor starts in a down state, the associated static routes are then removed from the IPv6 RIB and are reinserted when the BFDv6 session comes up. Therefore, you will see an interruption in traffic. This interruption can be avoided by configuring the static BFD neighbor as unassociated, waiting until the BFDv6 session has come up, and then reconfiguring the static BFD neighbor as associated.
- Transition from BFD monitoring of a static route—In this case, IPv6 static routes are monitored by BFD and inserted in the RIB. Here, you want to disable BFD monitoring of the static routes without interrupting traffic flow. This scenario can be achieved by first reconfiguring the static BFD neighbor as detached (thus disassociating the neighbor from the static routes) and then deconfiguring the static BFD neighbor.

BFD Support for OSPFv3

Bidirectional Forwarding Detection (BFD) supports OSPFv3.

How to Configure Bidirectional Forwarding Detection for IPv6

Specifying a Static BFDv6 Neighbor

An IPv6 static BFDv6 neighbor is specified separately from an IPv6 static route. An IPv6 static BFDv6 neighbor must be fully configured with the interface and neighbor address and must be directly attached to the local router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**] **Example:** `Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1`	Specifies static route IPv6 BFDv6 neighbors.

Associating an IPv6 Static Route with a BFDv6 Neighbor

IPv6 static routes are automatically associated with a static BFDv6 neighbor. A static neighbor is associated with a BFDv6 neighbor if the static next-hop explicitly matches the BFDv6 neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]
4. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]

Example:

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
```

Specifies static route BFDv6 neighbors.

Step 4 **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

Example:

```
Device(config)# ipv6 route 2001:DB8::/64 gigabitethernet 0/0/0 2001::1
```

Establishes static IPv6 routes.

Configuring BFD Support for OSPFv3

This section describes the procedures for configuring BFD support for OSPFv3, so that OSPFv3 is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPFv3 globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPFv3:

- You can enable BFD for all of the interfaces for which OSPFv3 is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ipv6 ospf bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPFv3 is routing by using the **ipv6 ospf bfd** command in interface configuration mode.

Note OSPF will only initiate BFD sessions for OSPF neighbors that are in the FULL state.

Configuring Baseline BFD Session Parameters on the Interface

Repeat this task for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Router(config-if)# bfd interval 50 min_rx 50 multiplier 5`	Enables BFD on the interface.

Configuring BFD Support for OSPFv3 for All Interfaces

Before you begin

OSPFv3 must be running on all participating devices. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id* [**vrf** *vpn-name*]
4. **bfd all-interfaces** [**strict-mode**]
5. **exit**
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# ipv6 router ospf 2`	Configures an OSPFv3 routing process.
Step 4	**bfd all-interfaces** [**strict-mode**] **Example:** `Device(config-router)# bfd all-interfaces`	Enables BFD for all interfaces participating in the routing process. **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Enter this command twice to go to privileged EXEC mode.
Step 6	**show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**] **Example:** `Device# show bfd neighbors detail`	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	**show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**] **Example:** `Device# show ipv6 ospf`	(Optional) Displays general information about OSPFv3 routing processes. If BFD is enabled in strict-mode, the command output displays `BFD is enabled in strict mode`.

Configuring BFDv6 Support for OSPFv3 on One or More OPSFv3 Interfaces

Before you begin

OSPFv3 must be running on all participating routers. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring Baseline BFD Session Parameters on the Interface, on page 82 section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ipv6 ospf bfd** [**disable**]
5. **exit**
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address*| **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 ospf bfd** [**disable**] **Example:** `Router(config-if)# ipv6 ospf bfd`	Enables BFD on a per-interface basis for one or more interfaces associated with the OSPFv3 routing process.
Step 5	**exit** **Example:** `Router(config-router)# exit`	Enter this command twice to go to privileged EXEC mode.
Step 6	**show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address*| **ipv6** *ipv6-address*] [**details**] **Example:** `Router# show bfd neighbors detail`	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	**show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**] **Example:** `Router# show ipv6 ospf`	(Optional) Displays general information about OSPFv3 routing processes.

Retrieving BFDv6 Information for Monitoring and Troubleshooting

SUMMARY STEPS

1. **enable**
2. **monitor event ipv6 static** [**enable** | **disable**]
3. **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**
4. **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**]
5. **debug ipv6 static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor event ipv6 static** [**enable**	**disable**] **Example:** `Device# monitor event ipv6 static enable`	Enables the use of event trace to monitor the operation of the IPv6 static and IPv6 static BFDv6 neighbors.
Step 3	**show ipv6 static** [*ipv6-address*	*ipv6-prefix/prefix-length*] [**interface** *type number*	**recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail** **Example:** `Device# show ipv6 static vrf vrf1 detail`	Displays the BFDv6 status for a static route associated with a static BFDv6 neighbor.
Step 4	**show ipv6 static** [*ipv6-address*	*ipv6-prefix/prefix-length*] [**interface** *type number*	**recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**] **Example:** `Device# show ipv6 static vrf vrf1 bfd`	Displays static BFDv6 neighbors and associated static routes.
Step 5	**debug ipv6 static** **Example:** `Device# debug ipv6 static`	Enables BFDv6 debugging.

Configuration Examples for Bidirectional Forwarding Detection for IPv6

Example: Specifying an IPv6 Static BFDv6 Neighbor

The following example specifies a fully configured IPv6 static BFDv6 neighbor. The interface is GigabitEthernet 0/0/0 and the neighbor address is 2001::1.

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
```

Example: Associating an IPv6 Static Route with a BFDv6 Neighbor

In this example, the IPv6 static route 2001:DB8::/32 is associated with the BFDv6 neighbor 2001::1 over the GigabitEthernet 0/0/0 interface:

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
Device(config)# ipv6 route 2001:DB8::/32 gigabitethernet 0/0/0 2001::1
```

Additional References

Related Documents

Related Topic	Document Title
OSPF for IPv6	"Implementing OSPF for IPv6," *Cisco IOS IPv6 Configuration Guide*
IPv6 static routes	"Implementing Static Routes for IPv6," *Cisco IOS IPv6 Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-bfd-v4v6-1hop-07.txt	*BFD for IPv4 and IPv6 (Single Hop)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing Bidirectional Forwarding for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for Implementing Bidirectional Forwarding for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: Static Route Support for BFD over IPv6	Cisco IOS XE Release 2.1	BFD for IPv6 is used to verify next-hop reachability for IPv6 static routes. The following commands were introduced or modified by this feature: **debug ipv6 static**, **ipv6 route**, **ipv6 route static bfd**, **monitor event ipv6 static**, **show ipv6 static**
OSPFv3 for BFD	Cisco IOS XE Release 2.1	BFD supports the dynamic routing protocol OSPF for IPv6 (OSPFv3). The following commands were introduced or modified by this feature: **bfd all-interfaces**, **bfd interval**, **ipv6 ospf bfd**, **ipv6 router ospf**, **show bfd neighbors**