CHAPTER 37

Configuring IPv6 Unicast Routing

Internet Protocol Version 6 (IPv6) is the network-layer Internet Protocol intended to replace Version 4 (IPv4) in the TCP/IP suite of protocols. This chapter describes how to configure IPv6 unicast routing on the Catalyst 3750 switch.

For information about configuring IPv4 unicast routing, see Chapter 36, "Configuring IP Unicast Routing." For information about configuring IPv6 Multicast Listener Discovery (MLD) snooping, see Chapter 38, "Configuring IPv6 MLD Snooping." For information on configuring IPv6 access control lists (ACLs), see Chapter 39, "Configuring IPv6 ACLs."

To use this feature, the stack master must be running the advanced IP services image, which is orderable from Cisco. This image includes all IP services image (formerly known as the enhanced multilayer image [EMI]) features plus IPv6 host and unicast routing support.

To enable IPv6 routing, you must configure a switch database management (SDM) template to a dual IPv4 and IPv6 template. See the "SDM Templates" section on page 37-18.

Unless otherwise noted, the term *switch* refers to a standalone switch and to a switch stack.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS documentation referenced in the procedures

This chapter consists of these sections:

- "Understanding IPv6" section on page 37-1
- "Configuring IPv6" section on page 37-20
- "Displaying IPv6" section on page 37-32

Understanding IPv6

The primary reason for using IPv6 is to increase Internet global address space to accommodate the rapidly increasing number of users and applications that require unique global IP addresses. IPv4 uses 32-bit addresses to provide approximately 4 billion available addresses. Large blocks of these addresses are allocated to government agencies and large organizations, and the number of available IP addresses is rapidly decreasing. IPv6 incorporates 128-bit source and destination addresses and can provide significantly more globally unique IP addresses than IPv4.

The architecture of IPv6 allows existing IPv4 users to transition easily to IPv6, and provides services such as end-to-end security, quality of service (QoS), and globally unique addresses. The flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT) processing by border routers at the edge of networks. IPv6 provides newer unicast methods, introduces hexadecimal values into the IP address, and uses colons (:) instead of periods (.) as delimiters.

IPv6 also provides these advantages over IPv4:

- Easier address management and delegation
- Easy address autoconfiguration with *stateless autoconfiguration*, which is similar to DHCP but does not require a specified DHCP application or server
- Embedded IPsec (encrypted security)
- Routing optimized for mobile devices
- Duplicate Address Detection (DAD) feature

For information about how Cisco Systems implements IPv6, go to this URL:

http://www.cisco.com//warp/public/732/Tech/ipv6/

This section describes IPv6 implementation on the switch. These sections are included:

- IPv6 Addresses, page 37-2
- Supported IPv6 Unicast Routing Features, page 37-3
- Unsupported IPv6 Unicast Routing Features, page 37-16
- Limitations, page 37-16
- IPv6 and Switch Stacks, page 37-17
- SDM Templates, page 37-18

IPv6 Addresses

IPv6 supports three types of addresses: unicast (one-to-one), multicast (one-to-many), and anycast (one-to-nearest). Multicast addresses replace the use of broadcast addresses. The switch supports only IPv6 unicast addresses. The switch does not support site-local unicast addresses, anycast addresses, or multicast addresses in this release.

The IPv6 128-bit addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons in the format: x:x:x:x:x:x:x:x. This is an example of an IPv6 address:

2031:0000:130F:0000:0000:09C0:080F:130B

For easier implementation, leading zeros in each field are optional. This is the same address without leading zeros:

2031:0:130F:0:0:9C0:80F:130B

You can also use two colons (::) to represent successive hexadecimal fields of zeros, but you can use this short version only once in each address:

2031:0:130F::09C0:080F:130B

For more information about IPv6 address formats, address types, and the IPv6 packet header, go to the "Implementing Addressing and Basic Connectivity" section of "The Cisco IOS IPv6 Configuration Library" at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

In the "Implementing Addressing and Basic Connectivity" chapter, these sections apply to the Catalyst 3750 switch:

- IPv6 Address Formats
- IPv6 Address Type: Unicast
- IPv6 Address Output Display
- Simplified IPv6 Packet Header

Supported IPv6 Unicast Routing Features

These sections describe the IPv6 protocol (RFC 2460) features supported by the switch:

- 128-Bit Wide Unicast Addresses, page 37-3
- Path MTU Discovery for IPv6 Unicast, page 37-4
- ICMPv6, page 37-4
- Neighbor Discovery, page 37-4
- IPv6 Stateless Autoconfiguration and Duplicate Address Detection, page 37-5
- IPv6 Applications, page 37-10
- Dual IPv4 and IPv6 Protocol Stacks, page 37-10
- EIGRP IPv6, page 37-11
- SNMP and Syslog Over IPv6, page 37-14
- HTTP(S) Over IPv6, page 37-15

Support on the switch includes expanded address capability, header format simplification, improved support of extensions and options, and hardware parsing of the extension header. The switch supports hop-by-hop extension header packets, which are routed or bridged in software.

The switch provides IPv6 routing capability over native Ethernet Inter-Switch Link (ISL) or 802.1Q trunk ports for static routes, Routing Information Protocol (RIP) for IPv6 (RFC 2080), and Open Shortest Path First (OSPF) Version 3 protocol (RFC 2740). It supports up to 16 equal-cost routes and can forward IPv4 and IPv6 frames simultaneously at line rate.

128-Bit Wide Unicast Addresses

The switch supports aggregatable global unicast addresses and link-local unicast addresses (RFC 2373). It does not support site-local unicast addresses.

- Aggregatable global unicast addresses are IPv6 addresses from the aggregatable global unicast prefix. The address structure enables strict aggregation of routing prefixes and limits the number of routing table entries in the global routing table. These addresses are used on links that are aggregated through organizations and eventually to the Internet service provider.

These addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Current global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). Addresses with a prefix of 2000::/3(001) through E000::/3(111) must have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format.

- Link local unicast addresses can be automatically configured on any interface by using the link-local prefix FE80::/10(1111 1110 10) and the interface identifier in the modified EUI format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes on a local link use link-local addresses and do not require globally unique addresses to communicate. IPv6 routers do not forward packets with link-local source or destination addresses to other links.

See the section on IPv6 Unicast Addresses in the "Implementing Addressing and Basic Connectivity for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

Each IPv6 host interface can support up to three addresses in hardware (one aggregatable global unicast address, one link-local unicast address, and zero or more privacy addresses).

DNS for IPv6

IPv6 introduces new Domain Name System (DNS) record types that are supported in the DNS name-to-address and address-to-name lookup processes. The new DNS AAAA resource record types support IPv6 addresses and are equivalent to an A address record in IPv4. The switch supports DNS resolution for IPv4 and IPv6.

Path MTU Discovery for IPv6 Unicast

The switch supports advertising the system MTU to IPv6 nodes and path MTU discovery. Path MTU discovery (RFC 1981) allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, if a link along the path is not large enough to accommodate the packet size, the source of the packet handles the fragmentation. The switch does not support path MTU discovery for multicast packets.

ICMPv6

The Internet Control Message Protocol (ICMP) in IPv6 (RFC 2463) functions the same as in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, to report errors during processing and other diagnostic functions. In IPv6, ICMP packets are also used in the neighbor discovery protocol and path MTU discovery. A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet.

Neighbor Discovery

The switch supports Neighbor Discovery Protocol (NDP) for IPv6 (RFC 2461), a protocol running on top of ICMPv6, and Static Neighbor Discovery for IPv6 stations that do not support NDP. 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 the neighbor, and keep track of neighboring routers.

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. These messages are sent on the local link when a node needs to determine the link-layer address of another node on the same local link. When a destination node receives a neighbor solicitation message, it replies by sending a neighbor advertisement message, which has a value of 136 in the ICMP packet header Type field.

A value of 137 in the ICMP packet header Type field identifies an IPv6 neighbor redirect message. The switch supports ICMPv6 redirect (RFC 2463) for routes with mask lengths less than 64. ICMP redirect is not supported for host routes or for summarized routes with mask lengths greater than 64. Routers send neighbor-redirect messages to inform hosts of better first-hop nodes on the path to a destination. A router does not update its routing tables after receiving a neighbor-redirect message and hosts do not originate neighbor-redirect messages.

Neighbor discovery throttling ensures that the switch CPU is not unnecessarily burdened while it is in the process of obtaining the next hop forwarding information to route an IPv6 packet. The switch performs a drop in hardware of any additional IPv6 packets whose next hop is the same neighbor the CPU is actively resolving. Performing this drop avoids adding further load on the CPU and results in a more efficient use of the switch CPU in an IPv6 routed environment.

IPv6 Stateless Autoconfiguration and Duplicate Address Detection

IPv6 supports two types of autoconfiguration:

- Stateless autoconfiguration where a host autonomously configures its own link-local address, and booting nodes send router solicitations to request router advertisements for configuring interfaces.
- Stateful autoconfiguration using Dynamic Host Configuration Protocol (DHCP) IPv6.

The switch supports stateless autoconfiguration to manage link, subnet, and site addressing changes, such as management of host and mobile IP addresses.

Every interface on IPv6 nodes must have a link-local address that is automatically configured from the identifier (router MAC address) 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 the help of a server such as a DHCP server.

With IPv6, a router on the link uses router advertisement messages to advertise global prefixes and its ability to act as a default router for the link. 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 router advertisement messages.

The 128-bit IPv6 addresses configured by a node are then subjected to duplicate-address detection to ensure their uniqueness on the link. If the advertised prefixes are globally unique, the IPv6 addresses configured by the node are guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the ICMP packet header Type field, are sent by hosts at system startup so that the host can be immediately autoconfigured without waiting for the next scheduled router advertisement message.

IPv6 duplicate-address detection is performed on unicast addresses before they are assigned to an interface. The switch does not support automatically generated site-local IPv6 addresses.

IPv6 Stateless Autoconfiguration

When an IPv6 host (nonrouter) autoconfigures its interfaces, the process includes generating a link-local, a site-local, and a global address through stateless address autoconfiguration.

IPv6 nodes (routers and hosts) begin the autoconfiguration process by generating a link-local address for the interface. Link-local address autoconfiguration is started by:

- Enabling IPv6 on an interface by entering the **ipv6 enable** interface configuration command
- Manually configuring the IPv6 address
- Autoconfiguring by entering the **ipv6 address autoconfig** command

A link-local address is formed by appending the interface identifier to the well-known link-local prefix (FE80::/10). The IPv6 node verifies that the generated tentative address is not used by another node on the link before the address can be assigned to the interface. To verify this, the IPv6 node sends a neighbor solicitation with the tentative address as the target address. If another node is detected to be using that address or is attempting to use that address (duplicate address detection), the node sends a neighbor solicitation for the target as well. If the tentative link-local address is not available, autoconfiguration stops and you must manually configure the interface.

IPv6 nodes with a 48-bit MAC address generate an identifier for the autoconfigured address by inserting 0xFF and 0xFE in the MAC address and reversing the universal/local bit. For example, if an interface MAC address is 000b.462e.9047, the identifier would be 020b:46ff:fe2e:9047, and the autogenerated IPv6 ink address would be FE80::20B:46FF:FE2E:9047.

Only IPv6 hosts can autoconfigure stateless addresses of site-local and global addresses that are started by using **ipv6 address autoconfig** on an interface. An IPv6 host sends router solicitations to the all-routers multicast group to obtain router advertisements.

IPv6 routers also periodically send router advertisements, but the delay between successive advertisements is generally a longer duration than for what a host performing autoconfiguration will wait. Router advertisements contain zero or more prefix information options that contain information that the stateless address autoconfiguration uses to generate site-local and global addresses.

Prefix information options specify the prefixes that are on-link and are used for address autoconfiguration. A router includes all of its on-link prefixes (except the link-local prefix) so that multihome hosts have complete prefix information about on-link destinations for the links to which they attach.

The autogenerated address is obtained by appending the interface IEEE EUI-64 to the prefix in the prefix information option in the router advertisement. If the sum of the prefix length and interface identifier length does not equal 128 bits, the prefix information option is ignored.

Link-local Address Configuration

These are examples of the **show** command output with **ipv6 enable** configured on an interface:

```
switch# show running-config interface fastethernet1/0/16
Building configuration...

Current configuration : 79 bytes
!
interface FastEthernet1/0/16
 no switchport
 no ip address
 ipv6 enable
end

switch# show interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up (connected)
Hardware is Fast Ethernet, address is 000b.462e.9047 (bia 000b.462e.9047)
```

```
switch# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  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.
```

Global-Address Configuration

This configuration shows site-local and global-address autoconfiguration enabled after using the **ipv6 address autoconfig** command on an interface:

```
switch# show running-config interface fastethernet1/0/16
Building configuration...

Current configuration : 104 bytes
!
interface FastEthernet1/0/16
 no switchport
 no ip address
 ipv6 address autoconfig
end
```

This is the configuration on the router side:

```
switch2# show running-config int gigabitethernet1/0/16
Building configuration...

Current configuration : 110 bytes
!
interface GigabitEthernet1/0/16
 no switchport
 no ip address
 no keepalive
 ipv6 address 1016:1::1/64
end

switch# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  Global unicast address(es):
    1016:1::20B:46FF:FE2E:9047, subnet is 1016:1::/64 [PRE]
      valid lifetime 2591958 preferred lifetime 604758
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047
```

This is the configuration with another address configured on the router:

```
switch2# show running-config interface gigabitethernet1/0/16
Building configuration...

Current configuration : 137 bytes
!
interface GigabitEthernet1/0/16
 no switchport
 no ip address
 no keepalive
 ipv6 address 1016:1::1/64
 ipv6 address 1016:2::1/64
end

switch2# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  Global unicast address(es):
    1016:1::20B:46FF:FE2E:9047, subnet is 1016:1::/64 [PRE]
      valid lifetime 2591998 preferred lifetime 604798
    1016:2::20B:46FF:FE2E:9047, subnet is 1016:2::/64 [PRE]
      valid lifetime 2591998 preferred lifetime 604798
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047

switch2# show running-config internet gigabitethernet1/0/16
Building configuration...

Current configuration : 137 bytes
!
interface GigabitEthernet1/0/16
 no switchport
 no ip address
 no keepalive
 ipv6 address 1016:1::1/64
 ipv6 address 1016:2::1/72
end

switch2# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  Global unicast address(es):
    1016:1::20B:46FF:FE2E:9047, subnet is 1016:1::/64 [PRE]
      valid lifetime 2591906 preferred lifetime 604706
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047
```

This is the configuration with Valid Lifetime and Preferred Lifetime with a prefix configured to nondefault values on the router side:

```
switch2# show running-config interface gigabitethernet1/0/16
Building configuration...

Current configuration : 173 bytes
!
interface GigabitEthernet1/0/16
 no switchport
 no ip address
 no keepalive
 ipv6 address 1016:1::1/64
 ipv6 address 1016:2::1/64
 ipv6 nd prefix 1016:2::/64 180 180
end

switch2# show running-config interface fastethernet1/0/16
Building configuration...

Current configuration : 91 bytes
!
interface FastEthernet1/0/16
 no switchport
 no ip address
 ipv6 address autoconfig
end

switch2# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  Global unicast address(es):
    1016:1::20B:46FF:FE2E:9047, subnet is 1016:1::/64 [PRE]
      valid lifetime 2591999 preferred lifetime 604799
    1016:2::20B:46FF:FE2E:9047, subnet is 1016:2::/64 [PRE/TEN]
      valid lifetime 179 preferred lifetime 179
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  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.
```

This is a site-local address configuration:

```
switch# show running-config interface fastethernet1/0/16
Building configuration...

Current configuration : 91 bytes
!
interface FastEthernet1/0/16
 no switchport
 no ip address
 ipv6 address autoconfig
end
```

This is the configuration on the router side:

```
switch# show running-config interface gigabitethernet1/0/16
Building configuration...

Current configuration : 115 bytes
!
interface GigabitEthernet1/0/16
 no switchport
 no ip address
 no keepalive
 ipv6 address FEC0:1016:1::1/64
end

switch# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
  Global unicast address(es):
    FEC0:1016:1:0:20B:46FF:FE2E:9047, subnet is FEC0:1016:1::/64 [PRE]
      valid lifetime 2591834 preferred lifetime 604634
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2E:9047
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  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.
```

IPv6 Applications

The switch has IPv6 support for these applications:

- Ping, traceroute, Telnet, TFTP, and FTP
- Secure Shell (SSH) over an IPv6 transport
- HTTP server access over IPv6 transport
- DNS resolver for AAAA over IPv4 transport
- Cisco Discovery Protocol (CDP) support for IPv6 addresses

For more information about managing these applications with Cisco IOS, see the “Managing Cisco IOS Applications over IPv6” section in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

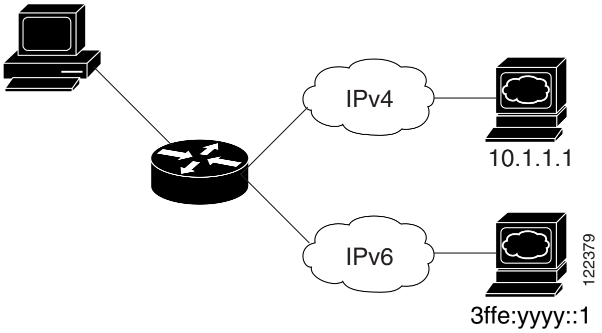
Dual IPv4 and IPv6 Protocol Stacks

One technique for transitioning to IPv6 is by using dual IPv4 and IPv6 protocol stacks. Using dual stacks enables gradual, one-by-one upgrades to applications running on nodes. Applications that are upgraded to IPv6 use the IPv6 protocol stack, and applications that are not upgraded and support only IPv4 can coexist with upgraded applications on the same node. New and upgraded applications can use both IPv4 and IPv6 protocol stacks.

The Cisco IOS software supports the dual IPv4 and IPv6 protocol stack technique. When both IPv4 and IPv6 routing are enabled and an interface is configured with both an IPv4 and IPv6 address, the interface forwards both IPv4 and IPv6 traffic.

Figure 37-1 shows a router forwarding both IPv4 and IPv6 traffic through the same interface, based on the IP packet and destination addresses.

Figure 37-1 Dual IPv4 and IPv6 Support on an Interface



The switch uses ternary content addressable memory (TCAM) to store unicast routes, MAC addresses, access control lists (ACLs), and other features, and provides the switch database management (SDM) templates to allocate memory resources depending on how the switch is used. You must use the dual IPv4 and IPv6 template templates to allocate TCAM usage to both IPv4and IPv6 protocols. See the "SDM Templates" section on page 37-18.

EIGRP IPv6

Enhanced Interior Gateway Routing Protocol (EIGRP) IPv6 is supported beginning with Cisco IOS Release 12.3(40)SE. The IPv6 commands duplicate the functionality of their counterpart commands for IPv4. EIGRP IPv6 can be configured directly onto the interfaces on which it runs and therefore does not require a global IPv6 address.

This example below shows the minimal configuration required for EIGRP IPv6:

```
!
ipv6 unicast-routing        enable IPv6 packet forwarding
!
interface e0
  ipv6 enable               enable IPv6 on the interface (or configure global address)
  ipv6 eigrp 1              enable EIGRP IPv6 for AS 1 (Nvgen's router process)
  no shutdown
!
ipv6 router eigrp 1         router process for AS 1
  router-id 1.1.1.1;        A explicit router id on this IPv6-only node
  no shutdown;              process is not shutdown
```

The above configuration allows you to send hellos to establish adjacencies (on Ethernet 0), however, to do so, it must also meet the following conditions:

- IPv6 unicast packet forwarding is enabled.
- IPv6 is enabled by issuing the **ipv6 enable** command on the interface or on a global IPv6 address.
- The interface has its line protocol running.
- The router process is running.
- The router process has a router ID.

EIGRP IPv6 can be configured on an interface. However, after being configured on an interface, it can start running before any of the EIGRP IPv6 router-mode commands have been executed. This is undesired, therefore, by default, EIGRP Ipv6 is in shutdown state. After you configure the router and interface for EIGRP IPv6, then you can issue the **no shutdown** command to start the protocol. This ensures that all router-mode configurations are executed before the protocol is started.

Prefix Lists

Use the **distribute-list prefix-list** command to define which networks running EIGRP IPv6 are to receive routing updates. The **route-map** command is not supported for route filtering with a distributed list.

Router ID

An instance of EIGRP IPv6 requires that you have a router ID before it can run. As with IPv4, EIGRP IPv6 supports implicit and explicit router IDs. An implicit router ID is derived from a local IPv4 address, so any IPv4 node always has an available router ID. However, EIGRP IPv6 might be running in a network with only IPv6 nodes and therefore might not have an available IPv4 router ID. You can use the **show ipv6 eigrp** command to see whether a router ID has been configured, and you can use the **router-id** command to set an explicit router ID.

Passive Interfaces

As with EIGRP IPv4, EIGRP IPv6 allows you to specify your EIGRP IPv4 interfaces and to select a subset of those to be passive interfaces. Use the **passive-interface default** command to make all interfaces passive, then use the **no passive-interface** command on selected interfaces to make them active. EIGRP IPv6 does not need to be configured on a passive interface.

EIGRP IPv6 Commands

Table 37-1 shows the supported interface commands for EIGRP IPv6 on the switch.

Table 37-1 EIGRP IPv6 Interface Commands for EIGRP IPv6

Command	Purpose
ipv6 eigrp *as-number*	Enables EIGRP IPv6 on a specified interface.
ipv6 authentication key-chain eigrp 1 *name*	Enables authentication of EIGRP IPv6 packets.
ipv6 authentication mode eigrp *as-number* **md5**	Specifies the type of authentication used in EIGRP IPv6 packets.
ipv6 bandwidth-percent eigrp *as-number percent*	Configures the percentage of bandwidth that may be used by EIGRP IPv6 on an interface.
ipv6 hello-interval eigrp *as-number seconds*	Configures the hello interval for the EIGRP IPv6 routing process designated by an autonomous system number.

Table 37-1 EIGRP IPv6 Interface Commands for EIGRP IPv6

Command	Purpose
ipv6 hold-time eigrp *as-number seconds*	Configures the hold time for a particular EIGRP IPv6 routing process designated by the autonomous system number.
ipv6 next-hop-self eigrp *as-number*	Informs the EIGRP IPv6 router that the IPv6 next hop is itself.
ipv6 split-horizon eigrp *as-number*	Enables EIGRP IPv6 split horizon.
ipv6 summary-address eigrp *as-number ipv6-prefix* [*admin-distance*]	Configures a summary aggregate address for a specified interface.

For more complete syntax and usage information on these commands, see the IOS see the IOS Command Reference publications.

Table 37-2 shows the supported router configuration commands for EIGRP IPv6 on the switch.

Table 37-2 EIGRP IPv6 Router-mode Commands

Command	Purpose
ipv6 router eigrp *as-number*	Specifies the EIGRP IPv6 routing process to be configured.
default-metric *bandwidth delay reliability loading mtu*	Sets metrics for EIGRP IPv6.
distance *internal-distance external-distance*	Allows the use of the best route to a node—either the internal or external distance.
distribute-list prefix-list *list-name*	Applies a prefix list to EIGRP IPv6 routing updates that are received or sent on an interface.
log-neighbor-changes	Enables the logging of changes in EIGRP IPv6 neighbor adjacencies.
log-neighbor-warnings *seconds*	Enables the logging of EIGRP neighbor warning messages.
maximum-paths *number-paths*	Controls the maximum number of equal-cost routes that is supported by a routing process for EIGRP IPv6.
metric weights *tos k1 k2 k3 k4 k5*	Tunes EIGRP metric calculations.
neighbor *x:x:x:x::x interface-name*	Defines a neighboring router with which to exchange routing information on a router that is running EIGRP.
passive-interface [*interface-name* | **default**]	Disables sending routing updates on an interface.
redistribute source-protocol [*process-id*] [*include-connected*] [*target-protocol-options*] [*source-protocol-options*]	Redistributes IPv6 routes from one routing domain into another routing domain.
router-id {*ip-address* | *ipv6-address*}	Uses a fixed router ID, use the router-id command in router configuration mode.
shutdown	Allows the user to shutdown the EIGRP protocol. The no version of this command allows the user to start EIGRP IPv6 protocol without changing any per-interface configuration.
stub [**connected** | **receive-only** | **static** | **summary** | **redistributed**]	Configures a router as a stub using EIGRP.
timers active-time [*time-limit* | **disabled**]	Adjusts routing wait time.
variance *multiplier*	Controls load balancing in an internetwork based on EIGRP.

For more complete syntax and usage information on these commands, see the IOS see the IOS Command Reference publications.

EIPRP of IPv6 supports the existing protocol-independent debug and show commands. Table 37-3 shows the supported show, debug, and clear commands for EIGRP IPv6.

Table 37-3 EIGRP IPv6 Show and Debug Commands

Command	Purpose
show ipv6 eigrp [*as-number*] *interface*	Displays information about interfaces configured for EIGRP IPv6.
show ipv6 eigrp [*as-number*] *neighbor*	Displays the neighbors discovered by EIGRP IPv6.
show ipv6 eigrp [*as-number*] *traffic*	Displays the number of EIGRP IPv6 packets sent and received.
show ipv6 eigrp topology [*as-number* | *ipv6-address*] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]	Displays entries in EIGRP of IPv6 IPv6 topology table.
debug ipv6 eigrp [*as-number*] [**neighbor** *ipv6-address* | **notification** | **summary**]	Displays information about EIGRP IPv6 protocol.
debug eigrp fsm	Displays debugging information about EIGRP feasible successor metrics (FSMs).
debug eigrp neighbors	Displays neighbors discovered by EIGRP.
debug eigrp packet	Displays debugging information for EIGRP IPv6 packets.
debug eigrp transmit	Displays transmittal messages sent by EIGRP.
debug eigrp nsf	Displays NSF notifications and information about NSF events in an EIGRP network.
clear ipv6 eigrp [**as-number**] [**neighbor** *ipv6-address*]	Deletes entries from EIGRP IPv6 routing tables.

For complete syntax and usage information on these commands, see the IOS see the IOS Command Reference publications.

SNMP and Syslog Over IPv6

Simple Network and Management Protocol (SNMP) can be configured over IPv6 transport so that an IPv6 host can send SNMP queries and receive SNMP notifications from a device running IPv6.

SNMP and syslog over IPv6 provides these features:

- Supports both IPv4 and IPv6
- Defines IPv6 transport for SNMP and modify the SNMP agent to support traps for an IPv6 host
- Enhances related MIBs to support IPv6 addressing scheme: modifies SNMP infra-MIBs that have IPv4 addresses so that they work with IPv6 addresses
- Configures IPv6 hosts as trap receiver

To provide support over IPv6, SNMP modifies existing IP transport mapping to support IPv4 and IPv6 simultaneously. To provide support for IPv6 transport mapping, SNMP provides these features:

- Opens User Datagram Protocol (UDP) SNMP socket with default settings
- Provides a new transport mechanism called, *SR_IPV6_TRANSPORT*
- Sends SNMP notifications over IPv6 transport

- Supports SNMP-named access lists for IPv6 transport
- Supports SNMP proxy forwarding using IPv6 transport
- Verifies SNMP Manger feature works with IPv6 transport

For information on SNMP over IPv6, see:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_mgev6.htm#wp1082849

For information on configuring an interface to support the IPv4 and IPv6 protocol stacks, see:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_configuration_guide_chapter09186a00806f3a6a.html#wp1060846

Syslog Over IPv6

To support both IPv4 and IPv6, management of IPv6 networks requires both IPv6 and IPv4 transports.

Syslog over IPv6 is responsible for transporting Cisco IOS-generated system error messages to configured servers. Syslog configures the connection to the logging host by using a Cisco IOS socket interface and starts a socket connection on the UDP or TCP transport by using Cisco IOS sockets.

Syslog supports common address data types that support both IPv4 and IPv6 transports. The syslog supports socket structures and APIs based on the user's CLI configurations. Socket structures and APIs support both IPv4 and IPv6 sockets.

The user specifies an IPv4-based logging host (syslog server) by using a host IP address in IPv4 format (i.e. 198.133.219.25). The user also specifies the TCP or UDP transport by using the IPv6 address (for example, ABCD:088A:EF75:1774::FFFF) of a logging host (syslog server) that supports IPv6 transport.

For syslog, IPv4 requires a 32-bit. IPv6 requires a 128-bit address.

Commands

These commands were enhanced for SNMP over IPV6. See the *Cisco IOS Command Reference Guide* for more information on these commands:

- **snmp-server host**
- **snmp-server community**
- **snmp-server engineID remote**
- **snmp-server group**
- **snmp-server user**
- **snmp mib target list**

HTTP(S) Over IPv6

The HTTP client in Cisco IOS supports sending requests to both IPv4 and IPv6 HTTP servers. The HTTP server in Cisco IOS can service HTTP requests from both IPv4 and IPv6 HTTP clients. URLs with literal IPv6 addresses must be formatted by using the rules listed in RFC 2732.

The accept socket call chooses an IPv4 or IPv6 address family according to protocol. The accept socket is either an IPv4 or IPv6 socket. The listening socket continues to listen for both IPv4 and IPv6 signals indicating a connection. The IPv6 listening socket is bound to an IPv6 wildcard address.

The underlying TCP/IP stack supports a dual-stack environment. HTTP relies on the TCP/IP stack and the sockets for processing network-layer interactions.

Basic network connectivity (**ping**) must exist between the client and the server hosts before HTTP connections can be made.

The output of these **show** commands can contain both IPv4 and IPv6 address types:

- **show ip http server history**
- **show ip http server connection**
- **show ip http client connection**
- **show ip http client history**

Unsupported IPv6 Unicast Routing Features

The switch does not support these IPv6 features in this release:

- IPv6 policy-based routing
- IPv6 virtual private network (VPN) routing and forwarding (VRF) table support
- Support for these IPv6 routing protocols: Multiprotocol Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS) routing, Enhanced Interior Gateway Routing Protocol (EIGRP)
- Simple Network Management Protocol (SNMP) over IPv6 transport
- IPv6 Hot Standby Router Protocol (HSRP)
- DHCPv6
- IPv6 packets destined to site-local addresses
- Tunneling protocols, such as IPv4-to-IPv6 or IPv6-to-IPv4
- The switch as a tunnel endpoint supporting IPv4-to-IPv6 or IPv6-to-IPv4 tunneling protocols
- IPv6 unicast reverse-path forwarding
- IPv6 general prefixes

Limitations

Because IPv6 is implemented in hardware in the switch, some limitations occur due to the use of IPv6 compressed addresses in the TCAM. These hardware limitations result in some loss of functionality and limits some features.

These are feature limitations.

- ICMPv6 redirect functionality is not supported for IPv6 host routes (routes used to reach a specific host) or for IPv6 routes with masks greater than 64. The switch cannot redirect hosts to a better first-hop router for a specific destination that is reachable through a host route or through a route with masks greater than 64.
- Load balancing using equal cost and unequal cost routes is not supported for IPv6 host routes or for IPv6 routes with a mask greater than 64.

- The switch cannot correctly forward SNAP-encapsulated IPv6 packets. These packets are corrupted before being forwarded (bridged or routed) and reach the network as corrupted packets.

Note There is a similar limitation for IPv4 SNAP-encapsulated packets, but the packets are dropped at the switch and are not forwarded as corrupted packets.

- The switch routes IPv6-to-IPv4 and IPv4-to-IPv6 packets in hardware, but the switch cannot be an IPv6-to-IPv4 or IPv4-to-IPv6 tunnel endpoint.
- Bridged IPv6 packets with hop-by-hop extension headers are forwarded in software. In IPv4, these packets are routed in software, but bridged in hardware.
- In addition to the normal SPAN and RSPAN limitations defined in the software configuration guide, these limitations are specific to IPv6 packets:
  - When you egress RSPAN IPv6-routed packets, the source MAC address in the SPAN output packet can be corrupted.
  - When you egress RSPAN IPv6-routed packets, the destination MAC address can be corrupted. Normal traffic is not affected.
- The switch cannot apply QoS classification, or policy-based routing on source-routed IPv6 packets in hardware.
- The switch cannot generate ICMPv6 `Packet Too Big` messages for multicast packets.

IPv6 and Switch Stacks

The switch supports IPv6 forwarding across the stack much the same as with IPv4 unicast routing. The stack master runs the IPv6 unicast routing protocols and computes the routing tables. Using distributed CEF (dCEF), the stack master downloads the routing table to the stack member switches. The member switches receive the tables and install IPv6 routes into hardware for hardware forwarding.

Note To route IPv6 packets in a stack, all switches in the stack should be running the advanced IP services image.

If a new switch becomes the stack master, the new master recomputes the IPv6 routing tables and distributes them to the member switches. While the new stack master is elected and is resetting, the switch stack does not forward IPv6 packets. If a new switch becomes the stack master, the stack MAC address also changes. When the IPv6 address of the stack is specified with an extended universal identifier (EUI) by using the **ipv6 address** *ipv6-prefix/prefix length* **eui-64** interface configuration command, the address is based on the interface MAC address, and changing the MAC address changes the IPv6 address. See the "Configuring IPv6 Addressing and Enabling IPv6 Routing" section on page 37-21.

Note If you configure the persistent MAC address feature on the stack and the stack master changes, the stack MAC address does not change for approximately four minutes. If the previous stack master rejoins the stack as a member switch during that time period, the stack MAC address remains the MAC address of the previous stack master. See the "Enabling Persistent MAC Address" section on page 5-19 in Chapter 5, "Managing Switch Stacks."

These are the functions of IPv6 stack master and members:

- Stack master:
  - runs IPv6 routing protocols
  - generates routing tables
  - distributes CEFv6 routing tables to stack members that use dCEFv6
  - runs IPv6 host functionality and IPv6 applications
- Stack member (must be running the advanced IP services image):
  - receives CEFv6 routing tables from the stack master
  - programs the routes into hardware

Note IPv6 packets are routed in hardware across the stack provided the packet does not have exceptions (IPv6Options) and the switches in the stack have not run out of hardware resources.

  - flushes the CEFv6 tables on master re-election

With IPv4 unicast routing, if the stack detects that the stack master is down and elects one of the stack members to be the new stack master, except for a momentary interruption, the hardware continues to forward packets with no protocols active. With IPv6, the switch does not continue forwarding packets. On election of a new stack master, the stack might need up to 60 seconds to recover all routes and resume forwarding traffic.

IPv6 host functionality is supported on the stack master, and all IPv6 applications run on the stack master.

SDM Templates

Most Catalyst 3750 switches have one TCAM to store unicast routes, MAC addresses, ACLs, and other features. To allocate TCAM resources for different usages, the switch SDM templates prioritize system resources to optimize support for certain features. You select the template that best suits the switch environment by entering the **sdm prefer** global configuration command. For more information about SDM templates, see Chapter 8, “Configuring SDM Templates.”

Note Aggregator templates are only supported on Catalyst 3750-12S switches. All other Catalyst 3750 switches support only the desktop templates.

The dual desktop and aggregator IPv4 and IPv6 templates allow the switch to be used in dual stack environments (supporting both IPv4 and IPv6).

Note If you try to configure IPv6 without first selecting a dual IPv4 and IPv6 template, a warning message is generated.

- In IPv4-only environments, the switch routes IPv4 packets and applies IPv4 QoS and ACLs in hardware. IPv6 packets are not supported.
- In dual IPv4 and IPv6 environments, the switch routes both IPv4 and IPv6 packets and applies IPv4 QoS and ACLs in hardware.

- IPv6 QoS and ACLs are not supported in this release.

Note If you do not plan to use IPv6, do not use the dual stack template because this template results in less TCAM capacity for each resource.

Dual IPv4-and-IPv6 SDM Templates

These SDM templates support IPv4 and IPv6 environments:

Note This release does not support IPv6 multicast routing or QoS. This release does support IPv6 Multicast Listener Discovery (MLD) snooping.

- Desktop dual IPv4 and IPv6 default SDM template—supports Layer 2, multicast, routing, QoS, and ACLs for IPv4; and Layer 2, routing, and ACLs for IPv6 on the desktop switches (all Catalyst 3750 switches except Catalyst 3750-12S).
- Desktop dual IPv4 and IPv6 routing template—supports Layer 2, multicast, routing (including policy-based routing), QoS, and ACLs for IPv4; and Layer 2, routing, and ACLs for IPv6 on the desktop switches (all Catalyst 3750 switches except Catalyst 3750-12S).
- Desktop dual IPv4 and IPv6 VLAN template—supports basic Layer 2, multicast, QoS, and ACLs for IPv4, and basic Layer 2 and ACLs for IPv6 on thedesktopswitches.
- Aggregator dual IPv4 and IPv6 default template—supports Layer 2, multicast, routing, QoS, and ACLs for IPv4, and Layer 2 and routing for IPv6 on Catalyst 3750-12S aggregator switches.
- Aggregator dual IPv4 and IPv6 routing template—supports Layer 2, multicast, routing (including policy-based routing), QoS, and ACLs for IPv4; and Layer 2, routing, and ACLs for IPv6 on Catalyst 3750-12S aggregator switches.
- Aggregator dual IPv4 and IPv6 VLAN template—supports basic Layer 2, multicast, QoS, and ACLs for IPv4,and basic Layer 2 and ACLs for IPv6 on Catalyst 3750-12S switches.

Note An IPv4 route requires only one TCAM entry. Because of the hardware compression scheme used for IPv6, an IPv6 route can take more than one TCAM entry, reducing the number of entries forwarded in hardware.

Table 37-4 defines the approximate feature resources allocated by each new template. Template estimations are based on a switch with eight routed interfaces and approximately one thousand VLANs.

Table 37-4 Approximate Feature Resources Allowed by Dual IPv4-IPv6 Templates

Resource	Desktop Default	Desktop Routing	Desktop VLAN	Aggregator Default	Aggregator Routing	Aggregator VLAN
Unicast MAC addresses	2 K	1536	8 K	2 K	2K	8 K
IPv4 IGMP groups and multicast routes	1 K	1K	1 K	2 K	2K	0
Total IPv4 unicast routes:	3 K	2816	0	3 K	8K	0
• Directly connected IPv4 hosts	2 K	1536	0	2 K	2K	0
• Indirect IPv4 routes	1 K	1280	0	1 K	6K	1 K

Table 37-4 Approximate Feature Resources Allowed by Dual IPv4-IPv6 Templates (continued)

Resource	Desktop Default	Desktop Routing	Desktop VLAN	Aggregator Default	Aggregator Routing	Aggregator VLAN
IPv6 multicast groups	1 K	1152	1 K	1 K	2176	1 K
Total IPv6 unicast routes:	3 K	2816	0	3 K	8K	0
• Directly connected IPv6 addresses	2 K	1536	0	2 K	2K	0
• Indirect IPv6 unicast routes	1 K	1280	0	1 K	6K	0
IPv4 policy-based routing ACEs	0	256	0	0	512	0
IPv4 or MAC QoS ACEs (total)	512	512	512	876	896	876
IPv4 or MAC security ACEs (total)	1 K	512	1K	512	1K	1 K
IPv6 policy-based routing ACEs[1]	0	255	0	0	510	0
IPv6 QoS ACEs	510	510	510	876	510	876
IPv6 security ACEs	510	510	510	876	510	876

[1] IPv6 policy-based routing is not supported in this release.

Configuring IPv6

These sections contain this IPv6 forwarding configuration information:

- Default IPv6 Configuration, page 37-20
- Configuring IPv6 Addressing and Enabling IPv6 Routing, page 37-21
- Configuring IPv4 and IPv6 Protocol Stacks, page 37-23
- Configuring IPv6 ICMP Rate Limiting, page 37-25
- Configuring CEF and dCEF for IPv6, page 37-25
- Configuring Static Routes for IPv6, page 37-26
- Configuring RIP for IPv6, page 37-28
- Configuring OSPF for IPv6, page 37-30

Default IPv6 Configuration

Table 37-5 shows the default IPv6 configuration.

Table 37-5 Default IPv6 Configuration

Feature	Default Setting
SDM template	Default desktop or default aggregator (Catalyst 3750-12S)
IPv6 routing	Disabled globally and on all interfaces

Table 37-5 Default IPv6 Configuration (continued)

Feature	Default Setting
CEFv6 or dCEFv6	Disabled (IPv4 CEF and dCEF are enabled by default) **Note** When IPv6 routing is enabled, CEFv6 and dCEF6 are automatically enabled.
IPv6 addresses	None configured

Configuring IPv6 Addressing and Enabling IPv6 Routing

This section describes how to assign IPv6 addresses to individual Layer 3 interfaces and enable the forwarding of IPv6 traffic globally on the switch.

Note In the **ipv6 address** interface configuration command, you must enter the *ipv6-address* and *ipv6-prefix* variables in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The *prefix-length* variable (preceded by a slash [/]) is a decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address).

For an interface to forward IPv6 traffic, you must configure an IPv6 address on the interface. Configuring a global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for the interface. The configured interface automatically joins these 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 (this address is used in the neighbor discovery process.)
- all-nodes link-local multicast group FF02::1
- all-routers link-local multicast group FF02::2

Note Before configuring IPv6 on the switch, be sure to select a dual IPv4 and IPv6 SDM template.

For more information about configuring IPv6 routing, see the "Implementing Addressing and Basic Connectivity for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

Note Not all features discussed in this chapter are supported by the Catalyst 3750 switch. See the "Unsupported IPv6 Unicast Routing Features" section on page 37-16.

Beginning in privileged EXEC mode, follow these steps to assign an IPv6 address to a Layer 3 interface and enable IPv6 routing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**sdm prefer dual-ipv4-and-ipv6** {**default** | **routing** | **vlan**} [**desktop**]	Select an SDM template that supports IPv4 and IPv6. • **default**—Set the switch to the default template to balance system resources. • **routing**—Set the switch to the routing template to support IPv4 and IPv6 routing, including IPv4 policy-based routing. • **vlan**—Maximize VLAN configuration on the switch with no routing supported in hardware. • **desktop**—Supported only on Catalyst 3750-12S aggregator switches to set the switch to one of the desktop templates. If not selected on an aggregator switch, an aggregator template is automatically selected.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**reload**	Reload the operating system.
Step 5	**configure terminal**	Enter global configuration mode (after the switch has reloaded).
Step 6	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure. The interface can be a physical interface, a switch virtual interface (SVI), or a Layer 3 EtherChannel.
Step 7	**no switchport**	Remove the interface from Layer 2 configuration mode (if it is a physical interface).
Step 8	**ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **ipv6 address** *ipv6-address* **link-local** or **ipv6 enable**	Specify a global IPv6 address with an extended universal identifier (EUI) in the low-order 64 bits of the IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. This enables IPv6 processing on the interface. Specify a link-local address on the interface to be used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on the interface. Automatically configure an IPv6 link-local address on the interface, and enable the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.
Step 9	**exit**	Return to global configuration mode.
Step 10	**ip routing**	Enable IP routing on the switch.
Step 11	**ipv6 unicast-routing**	Enable forwarding of IPv6 unicast data packets.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip v6 interface** *interface-id*	Verify your entries.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove an IPv6 address from an interface, use the **no ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **no ipv6 address** *ipv6-address* **link-local** interface configuration command. To remove all manually configured IPv6 addresses from an interface, use the **no ipv6 address** interface configuration command without arguments. To disable IPv6 processing on an interface that has not been explicitly configured with an IPv6 address, use the **no ipv6 enable** interface configuration command. To globally disable IPv6 routing, use the **no ipv6 unicast-routing** global configuration command.

This example shows how to enable IPv6 with both a link-local address and a global address based on the IPv6 prefix 2001:0DB8: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** EXEC command is included to show how the interface ID (20B:46FF:FE2F:D940) is appended to the link-local prefix FE80::/64 of the interface.

```
Switch(config)# sdm prefer dual-ipv4-and-ipv6 default
Switch(config)# ipv6 unicast-routing
Switch(config)# interface fastethernet1/0/11
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Switch(config-if)# end
Switch# show ipv6 interface fastethernet1/0/11
FastEthernet1/0/11 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
  2001:0DB8:c18:1:20B:46FF:FE2F:D940, subnet is 2001:0DB8:c18:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  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.
```

Configuring IPv4 and IPv6 Protocol Stacks

When you configure an interface with both an IPv4 and IPv6 address, the interface forwards both IPv4 and IPv6 traffic and can send and receive data on both IPv4 and IPv6 networks.

Beginning in privileged EXEC mode, follow these steps to configure a Layer 3 interface to support both the IPv4 and IPv6 and enable IPv6 routing.

Note Before configuring IPv6 routing, you must select an SDM template that supports IPv4 and IPv6. If not already configured, use the **sdm prefer dual-ipv4-and-ipv6** {**default** | **routing** | **vlan**} [**desktop**] global configuration command to configure a template that supports IPv6. When you select a new template, you must reload the switch by using the **reload** privileged EXEC command for the template to take effect.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip routing**	Enable routing on the switch.

	Command	Purpose
Step 3	**ipv6 unicast-routing**	Enable forwarding of IPv6 data packets on the switch.
Step 4	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 5	**no switchport**	Remove the interface from Layer 2 configuration mode (if it is a physical interface).
Step 6	**ip address** *ip-address mask* [**secondary**]	Specify a primary or secondary IPv4 address for the interface.
Step 7	**ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **ipv6 address** *ipv6-address* **link-local** or **ipv6 enable**	Specify a global IPv6 address with an interface identifier in the low-order 64 bits of the IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. This enables IPv6 processing on the interface. Specify a link-local address on the interface to be used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on the interface. Automatically configure an IPv6 link-local address on the interface, and enable the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show interface** *interface-id* **show ip interface** *interface-id* **show ip v6 interface** *interface-id*	Verify your entries.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable IPv4 routing, use the **no ip routing** global configuration command. To disable IPv6 routing, use the **no ipv6 unicast-routing** global configuration command. To remove an IPv4 address from an interface, use the **no ip address** *ip-address mask* interface configuration command. To remove an IPv6 address from an interface, use the **no ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **no ipv6 address** *ipv6-address* **link-local** interface configuration command. To remove all manually configured IPv6 addresses from an interface, use the **no ipv6 address** interface configuration command without arguments. To disable IPv6 processing on an interface that has not been explicitly configured with an IPv6 address, use the **no ipv6 enable** interface configuration command.

This example shows how to enable IPv4 and IPv6 routing on an interface.

```
Switch(config)# sdm prefer dual-ipv4-and-ipv6 default
Switch(config)# ip routing
Switch(config)# ipv6 unicast-routing
Switch(config)# interface fastethernet1/0/11
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.168.99.1 244.244.244.0
Switch(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Switch(config-if)# end
```

Configuring IPv6 ICMP Rate Limiting

IPv6 ICMP rate limiting uses a token-bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent to the network. The interval between error messages is specified in a time interval and a bucket size. Because some applications, such as traceroute, sometimes require replies to a group of requests to be sent out in rapid succession, specifying only the interval between error messages can cause the application to fail. The token bucket allows a number of tokens, each representing the ability to send one error message, to be stored in virtual buckets. For every 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, IPv6 ICMP error messages are not sent until a new token is placed in the bucket. This method does not increase the average rate-limiting time interval, but it provides more flexibility than fixed-time intervals.

ICMP rate limiting is enabled by default with a default interval between error messages of 100 milliseconds and a bucket size (maximum number of tokens to be stored in a bucket) of 10.

Beginning in privileged EXEC mode, follow these steps to change the ICMP rate-limiting parameters:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 icmp error-interval** *interval* [*bucketsize*]	Configure the interval and bucket size for IPv6 ICMP error messages: • *interval*—The interval (in milliseconds) between tokens being added to the bucket. The range is from 0 to 2147483647 milliseconds. • bucketsize—(Optional) The maximum number of tokens stored in the bucket. The range is from 1 to 200.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show ipv6 interface** [*interface-id*]	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default configuration, use the **no ipv6 icmp error-interval** global configuration command.

This example shows how to configure an IPv6 ICMP error message interval of 50 milliseconds and a bucket size of 20 tokens.

```
Switch(config)#ipv6 icmp error-interval 50 20
```

Configuring CEF and dCEF for IPv6

Cisco Express Forwarding (CEF) is a Layer 3 IP switching technology used to optimize network performance. CEF implements an advanced IP look-up and forwarding algorithm to deliver maximum Layer 3 switching performance. It is less CPU-intensive than fast-switching route-caching, allowing more CPU processing power to be dedicated to packet forwarding. In a Catalyst 3750 switch stack, the hardware uses distributed CEF (dCEF) in the stack. IPv4 CEF and dCEF are enabled by default. IPv4 CEF and dCEF are disabled by default, but automatically enabled when you configure IPv6 routing.

To route IPv6 unicast packets, you must first globally configure forwarding of IPv6 unicast packets by using the **ipv6 unicast-routing** global configuration command, and you must configure an IPv6 address and IPv6 processing on an interface by using the **ipv6 address** interface configuration command.

To disable IPv6 CEF or distributed CEF, use the **no ipv6 cef** or **no ipv6 cef distributed** global configuration command. To reenable IPv6 CEF or dCEF if it has been disabled, use the **ipv6 cef** or **ipv6 cef distributed** global configuration command. You can verify the IPv6 state by entering the **show ipv6 cef** privileged EXEC command.

Configuring Static Routes for IPv6

Static routes are manually configured and define an explicit route between two networking devices. The benefits of static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols because there is no requirement for routes to be calculated and communicated. The main disadvantage of using static routes is that static routes are not automatically updated, as with a dynamic routing protocol, and must be manually reconfigured if the network topology changes. Static routes are useful for smaller networks with only one path to an outside network or to provide security for a larger network for certain types of traffic.

There are types of static routes:

- Directly attached static routes—Only the output interface is specified because the destination is assumed to be directly attached to this interface. The packet destination is used as the next hop address. A directly attached static route is valid only when the specified interface is IPv6-enabled and is up.
- Recursive static routes—Only the next hop is specified, and the output interface is derived from the next hop. A recursive static route is valid only when the specified next hop results in a valid IPv6 output interface, the route does not self-recur, and the recursion depth does not exceed the maximum IPv6 forwarding recursion depth.
- Fully specified static routes—Both the output interface and the next hop are specified. The next hop is assumed to be directly attached to the specified output interface. A fully specified route is valid when the specified IPv6 interface is IPv6-enabled and up.
- Floating static routes—Any of the three types of static routes can be floating static routes, used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a less efficient administrative distance than the routing protocol it is backing up. Therefore, the dynamic route is always used for routing traffic in preference to the floating static route. If the dynamic route is lost, the floating static route is used in its place.

Note Before configuring a static IPv6 route, you must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on at least one Layer 3 interface by configuring an IPv6 address on the interface.

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 static route:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*]	Configure a static IPv6 route. • *ipv6-prefix*—The IPv6 network that is the destination of the static route. It can also be a hostname when static host routes are configured. • */prefix length*—The length of the IPv6 prefix. A decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. • *ipv6-address*—The IPv6 address of the next hop that can be used to reach the specified network. The IPv6 address of the next hop need not be directly connected; recursion is done to find the IPv6 address of the directly connected next hop. The address must be in the form documented in RFC 2373, specified in hexadecimal using 16-bit values between colons. • *interface-id*—Specify direct static routes from point-to-point and broadcast interfaces. With point-to-point interfaces, there is no need to specify the IPv6 address of the next hop. With broadcast interfaces, you should always specify the IPv6 address of the next hop, or ensure that the specified prefix is assigned to the link, specifying a link-local address as the next hop. You can optionally specify the IPv6 address of the next hop to which packets are sent. **Note** You must specify an *interface-id* when using a link-local address as the next hop (the link-local next hop must also be an adjacent router). • *administrative distance*—(Optional) An administrative distance. The range is 1 to 254; the default value is 1, which gives static routes precedence over any other type of route except connected routes. To configure a floating static route, use an administrative distance greater than that of the dynamic routing protocol.
Step 3	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 4	**show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix length*] [**interface** *interface-id*] [**recursive**] [**detail**] or **show ipv6 route static** [*updated*]	Verify your entries by displaying the contents of the IPv6 routing table. • **interface** *interface-id*—(Optional) Display only those static routes with the specified interface as an egress interface. • **recursive**—(Optional) Display only recursive static routes. The **recursive** keyword is mutually exclusive with the **interface** keyword, but it can be used with or without the IPv6 prefix included in the command syntax. • **detail**—(Optional) Display this additional information: – For valid recursive routes, the output path set, and maximum resolution depth. – For invalid routes, the reason why the route is not valid.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove a configured static route, use the **no ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] global configuration command.

This example shows how to configure a floating static route to an interface with an administrative distance of 130:

```
Switch(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130
```

For more information about configuring static IPv6 routing, see the "Implementing Static Routes for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

Configuring RIP for IPv6

Routing Information Protocol (RIP) for IPv6 is a distance-vector protocol that uses hop count as a routing metric. IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. IPv6 RIP enhancements include support for IPv6 addresses and prefixes and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages.

Each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB), that contains a set of best-cost IPv6 routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it stores only the lowest-cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors that are running RIP. If the same route is learned from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route is not added to the IPv6 RIB, but the route still exists in the IPv6 RIP RIB.

Note Before configuring the switch to run IPv6 RIP, you must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on any Layer 3 interfaces on which IPv6 RIP is to be enabled.

Beginning in privileged EXEC mode, follow these required and optional steps to configure IPv6 RIP:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 router rip** *name*	Configure an IPv6 RIP routing process, and enter router configuration mode for the process.
Step 3	**maximum-paths** *number-paths*	(Optional) Define the maximum number of equal-cost routes that IPv6 RIP can support. The range is from 1 to 64, and the default is four paths.
Step 4	**exit**	Return to global configuration mode.
Step 5	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 6	**ipv6 rip** *name* **enable**	Enable the specified IPv6 RIP routing process on the interface.
Step 7	**ipv6 rip** *name* **default-information** {**only** | **originate**}	(Optional) Originate the IPv6 default route (::/0) into the RIP routing process updates sent from the specified interface. **Note** To avoid routing loops after the IPv6 default route (::/0) is originated from any interface, the routing process ignores all default routes received on any interface. • **only**—Select to originate the default route, but suppress all other routes in the updates sent on this interface. • **originate**—Select to originate the default route in addition to all other routes in the updates sent on this interface.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show ipv6 rip** [*name*] [**interface** *interface-id*] [**database**] [**next-hops**] or **show ipv6 route rip** [*updated*]	Display information about current IPv6 RIP processes. Display the current contents of the IPv6 routing table.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable a RIP routing process, use the **no ipv6 router rip** *name* global configuration command. To disable the RIP routing process for an interface, use the **no ipv6 rip** *name* interface configuration command.

This example shows how to enable the RIP routing process *cisco*, with a maximum of eight equal-cost routes and enable it on an interface:

```
Switch(config)# ipv6 router rip cisco
Switch(config-router)# maximum-paths 8
Switch(config)# exit
Switch(config)# interface fastethernet2/0/11
Switch(config-if)# ipv6 rip cisco enable
```

For more information about configuring RIP routing for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

Configuring OSPF for IPv6

Open Shortest Path First (OSPF) is a link-state protocol for IP, which means that routing decisions are based on the states of the links that connect the source and destination devices. The state of a link is a description of the interface and its relationship to its neighboring networking devices. Interface information, which is propagated in link-state advertisements (LSAs), includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the routers connected to that network, and so on. LSA data is stored in a link-state database, which is used to create the OSPF routing table. While the database contains a collection of raw data, the routing table contains a list of shortest paths to known destinations that use specific Layer 3 ports. OSPF Version 2 (RFC 2740) supports IPv6.

OSPF for IPv6 is largely the same as OSPF Version 2 (for IPv4), expanded to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. However, there are these differences:

- Enabling OSPF for IPv6 on an interface automatically creates a routing process and its associated configuration; you do not need to explicitly create a routing process as in IPv4.
- In OSPF for IPv6, you must enable OSPF on each interface by using commands in interface configuration mode. In OSPF Version 2, interfaces are indirectly enabled by using router configuration mode.
- In IPv6, you can configure many address prefixes on an interface. All address prefixes configured on an interface are included by default; you cannot select a subset of address prefixes to import.
- Unlike OSPF Version 2, multiple instances of IPv6 can run on a link.
- OSPF Version 2 uses the 32-bit IPv4 address configured on the interface to chose an IPv4 address to use as the router ID. When you enable OSPF for IPv6 on an interface, if an IPv4 address is configured on the interface that IP address is used for the IPv6 router ID. If no IPv4 address is configured on the interface, you must use the **router-id** router configuration command to configure a router ID before the OSFP process is started.

OSPF automatically chooses a loopback interface over other interfaces and chooses the highest IP address among all loopback interfaces. If no loopback interfaces are present, OSPF selects the highest IP address in the router. You cannot configure OSPF to use any particular interface.

You can customize OSPF for IPv6 for your network, but you will most likely not need to. The defaults for OSPF in IPv6 are set to meet the requirements of most customers and features.

Note Be careful when changing the defaults for IPv6 commands. Changing the defaults might adversely affect OSPF for the IPv6 network.

Note Before you enable IPv6 OSPF on an interface, you must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on Layer 3 interfaces on which you are enabling IPv6 OSPF.

Beginning in privileged EXEC mode, follow these required and optional steps to configure IPv6 OSPF:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 router ospf** *process-id*	Enable OSPF router configuration mode for the process. The process ID is the number assigned administratively when enabling the OSPF for IPv6 routing process. It is locally assigned and can be a positive integer from 1 to 65535.
Step 3	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*]	(Optional) Consolidate and summarize routes at an area boundary. • *area-id*—Identifier of the area about which routes are to be summarized. It can be specified as either a decimal value or as an IPv6 prefix. • *ipv6-prefix/prefix length*—The destination IPv6 network and a decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark (/) must precede the decimal value. • **advertise**—(Optional) Set the address range status to advertise and generate a Type 3 summary link-state advertisement (LSA). • **not-advertise**—(Optional) Set the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and component networks remain hidden from other networks. • **cost** *cost*—(Optional) Metric or cost for this summary route, which is used during OSPF SPF calculation to determine the shortest paths to the destination. The value can be 0 to 16777215.
Step 4	**maximum paths** *number-paths*	(Optional) Define the maximum number of equal-cost routes to the same destination that IPv6 OSPF should enter in the routing table. The range is from 1 to 64, and the default is 16 paths.
Step 5	**exit**	Return to global configuration mode.
Step 6	**interface** *interface-id*	Enter interface configuration mode, and specify the Layer 3 interface to configure.
Step 7	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]	Enable OSPF for IPv6 on the interface. • **instance** *instance-id*—(Optional) Instance identifier.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-id*] or **show ipv6 ospf** [*process-id*] [*area-id*]	Display information about OSPF interfaces. Display general information about OSPF routing processes.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable an OSPF routing process, use the no **ipv6 router ospf** *process-id* global configuration command. To disable the OSPF routing process for an interface, use the **no ipv6 ospf** *process-id* **area** *area-id* interface configuration command.

For more information about configuring OSPF routing for IPv6, see the "Implementing OSPF for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00807fcf4b.html

Displaying IPv6

Table 37-6 shows the privileged EXEC commands for monitoring IPv6 on the switch.

Table 37-6 Commands for Monitoring IPv6

Command	Purpose
show ipv6 access-list	Display a summary of access lists.
show ipv6 cef	Display Cisco Express Forwarding for IPv6.
show ipv6 interface *interface-id*	Display IPv6 interface status and configuration.
show ipv6 mtu	Display IPv6 MTU per destination cache.
show ipv6 neighbors	Display IPv6 neighbor cache entries.
show ipv6 ospf	Display IPv6 OSPF information.
show ipv6 prefix-list	Display a list of IPv6 prefix lists.
show ipv6 protocols	Display IPv6 routing protocols on the switch.
show ipv6 rip	Display IPv6 RIP routing protocol status.
show ipv6 route	Display the IPv6 route table entries.
show ipv6 routers	Display the local IPv6 routers.
show ipv6 static	Display IPv6 static routes.
show ipv6 traffic	Display IPv6 traffic statistics.

This is an example of the output from the **show ipv6 interface** privileged EXEC command:

```
Switch# show ipv6 interface
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
    3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  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
<output truncated>
```

This is an example of the output from the **show ipv6 cef** privileged EXEC command:

```
Switch# show ipv6 cef
::/0
  nexthop 3FFE:C000:0:7::777 Vlan7
3FFE:C000:0:1::/64
  attached to Vlan1
3FFE:C000:0:1:20B:46FF:FE2F:D940/128
  receive
3FFE:C000:0:7::/64
  attached to Vlan7
3FFE:C000:0:7::777/128
  attached to Vlan7
3FFE:C000:0:7:20B:46FF:FE2F:D97F/128
  receive
3FFE:C000:111:1::/64
  attached to FastEthernet1/0/11
3FFE:C000:111:1:20B:46FF:FE2F:D945/128
  receive
3FFE:C000:168:1::/64
  attached to FastEthernet2/0/43
3FFE:C000:168:1:20B:46FF:FE2F:D94B/128
  receive
3FFE:C000:16A:1::/64
  attached to Loopback10
3FFE:C000:16A:1:20B:46FF:FE2F:D900/128
  receive

<output truncated>
```

This is an example of the output from the **show ipv6 protocols** privileged EXEC command:

```
Switch# show ipv6 protocols
IPv6 Routing Protocol is “connected”
IPv6 Routing Protocol is “static”
IPv6 Routing Protocol is “rip fer”
  Interfaces:
    Vlan6
    FastEthernet2/0/4
    FastEthernet2/0/11
    FastEthernet1/0/12
  Redistribution:
    None
```

This is an example of the output from the **show ipv6 rip** privileged EXEC command:

```
Switch# show ipv6 rip
RIP process "fer", port 521, multicast-group FF02::9, pid 190
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 9040, trigger updates 60
  Interfaces:
    Vlan6
    FastEthernet2/0/4
    FastEthernet2/0/11
    FastEthernet1/0/12
Redistribution:
    None
```

This is an example of the output from the **show ipv6 static** privileged EXEC command:

```
Switch# show ipv6 static
IPv6 Static routes
Code: * - installed in RIB
* ::/0 via nexthop 3FFE:C000:0:7::777, distance 1
```

This is an example of the output from the **show ipv6 neighbor** privileged EXEC command:

```
Switch# show ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
3FFE:C000:0:7::777                          - 0007.0007.0007  REACH Vl7
3FFE:C101:113:1::33                         - 0000.0000.0033  REACH Fa1/0/13
```

This is an example of the output from the **show ipv6 route** privileged EXEC command:

```
Switch# show ipv6 route
IPv6 Routing Table - Default - 1 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
L   FF00::/8 [0/0]
     via Null0, receive
```

This is an example of the output from the **show ipv6 traffic** privileged EXEC command.

```
Switch# show ipv6 traffic
IPv6 statistics:
  Rcvd:  1 total, 1 local destination
         0 source-routed, 0 truncated
         0 format errors, 0 hop count exceeded
         0 bad header, 0 unknown option, 0 bad source
         0 unknown protocol, 0 not a router
         0 fragments, 0 total reassembled
         0 reassembly timeouts, 0 reassembly failures
  Sent:  36861 generated, 0 forwarded
         0 fragmented into 0 fragments, 0 failed
         0 encapsulation failed, 0 no route, 0 too big
         0 RPF drops, 0 RPF suppressed drops
  Mcast: 1 received, 36861 sent

ICMP statistics:
  Rcvd: 1 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, 0 router advert, 0 redirects
        0 neighbor solicit, 0 neighbor advert
  Sent: 10112 output, 0 rate-limited
        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
        0 router solicit, 9944 router advert, 0 redirects
        84 neighbor solicit, 84 neighbor advert
```

```
UDP statistics:
  Rcvd: 0 input, 0 checksum errors, 0 length errors
        0 no port, 0 dropped
  Sent: 26749 output

TCP statistics:
  Rcvd: 0 input, 0 checksum errors
  Sent: 0 output, 0 retransmitted
```