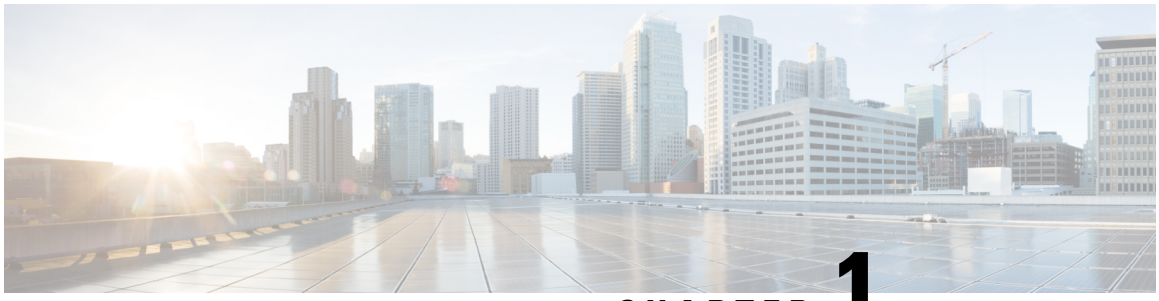
IPv6 Configuration Guide, Cisco IOS XE Gibraltar 16.10.x (Catalyst 9500 Switches)

First Published: 2018-11-15

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

Configuring IPv6 Unicast Routing

Information About Configuring IPv6 Unicast Routing

This chapter describes how to configure IPv6 unicast routing on the switch.

Note To use all IPv6 features in this chapter, the switch or active switch must be running the Network Advantage license. Switches running the Network Essentials license support IPv6 static routing and RIP for IPv6. Switches running the Network Advantage license support OSPF, EIGRP and BGP for IPv6.

Understanding IPv6

IPv4 users can move to IPv6 and receive services such as end-to-end security, quality of service (QoS), and globally unique addresses. The IPv6 address space reduces the need for private addresses and Network Address Translation (NAT) processing by border routers at network edges.

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

http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html

For information about IPv6 and other features in this chapter

- See the *Cisco IOS IPv6 Configuration Library*.
- Use the Search field on Cisco.com to locate the Cisco IOS software documentation. For example, if you want information about static routes, you can enter *Implementing Static Routes for IPv6* in the search field to learn about static routes.

IPv6 Addresses

The switch supports only IPv6 unicast addresses. It does not support site-local unicast addresses, or anycast addresses.

The IPv6 128-bit addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons in the format: n:n:n:n:n:n:n:n. 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, see the http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-3e/ip6b-xe-3e-book.html of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

In the "Information About Implementing Basic Connectivity for IPv6" chapter, these sections apply to the switch:

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

Supported IPv6 Unicast Routing Features

These sections describe the IPv6 protocol features supported by the switch:

The switch provides IPv6 routing capability over Routing Information Protocol (RIP) for IPv6, and Open Shortest Path First (OSPF) Version 3 Protocol. It supports up to 16 equal-cost routes and can simultaneously forward IPv4 and IPv6 frames at line rate.

128-Bit Wide Unicast Addresses

The switch supports aggregatable global unicast addresses and link-local unicast addresses. 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 unique 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 (NDP) 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.

For more information, see the section about IPv6 unicast addresses in the "Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

DNS for IPv6

IPv6 supports Domain Name System (DNS) record types in the DNS name-to-address and address-to-name lookup processes. The 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 maximum transmission unit (MTU) to IPv6 nodes and path MTU discovery. Path MTU discovery 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.

ICMPv6

The Internet Control Message Protocol (ICMP) in IPv6 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.

Neighbor Discovery

The switch supports NDP for IPv6, a protocol running on top of ICMPv6, and static neighbor entries 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), to verify the reachability of the neighbor, and to keep track of neighboring routers.

The switch supports ICMPv6 redirect for routes with mask lengths less than 64 bits. ICMP redirect is not supported for host routes or for summarized routes with mask lengths greater than 64 bits.

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 drops any additional IPv6 packets whose next hop is the same neighbor that the switch is actively trying to resolve. This drop avoids further load on the CPU.

Default Router Preference

The switch supports IPv6 default router preference (DRP), an extension in router advertisement messages. DRP improves the ability of a host to select an appropriate router, especially when the host is multihomed and the routers are on different links. The switch does not support the Route Information Option in RFC 4191.

An IPv6 host maintains a default router list from which it selects a router for traffic to offlink destinations. The selected router for a destination is then cached in the destination cache. NDP for IPv6 specifies that routers that are reachable or probably reachable are preferred over routers whose reachability is unknown or suspect. For reachable or probably reachable routers, NDP can either select the same router every time or cycle through the router list. By using DRP, you can configure an IPv6 host to prefer one router over another, provided both are reachable or probably reachable.

For configuring DRP for IPv6, see the *Configuring Default Router Preference* section.

For more information about DRP for IPv6, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Stateless Autoconfiguration and Duplicate Address Detection

The switch uses stateless autoconfiguration to manage link, subnet, and site addressing changes, such as management of host and mobile IP addresses. A host autonomously configures its own link-local address, and booting nodes send router solicitations to request router advertisements for configuring interfaces.

For more information about autoconfiguration and duplicate address detection, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Applications

The switch has IPv6 support for these applications:

- Ping, traceroute, Telnet, and TFTP
- FTP (This is supported only on Cisco Catalyst 9500 Series Switches - High Performance)
- 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, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

DHCP for IPv6 Address Assignment

DHCPv6 enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 clients. The address assignment feature manages non-duplicate address assignment in the correct prefix based on the network where the host is connected. Assigned addresses can be from one or multiple prefix pools. Additional options, such as default domain and DNS name-server address, can be passed back to the client. Address pools can be assigned for use on a specific interface, on multiple interfaces, or the server can automatically find the appropriate pool.

For configuring DHCP for IPv6, see the *Configuring DHCP for IPv6 Address Assignment* section.

For more information about configuring the DHCPv6 client, server, or relay agent functions, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Static Routes for IPv6

Static routes are manually configured and define an explicit route between two networking devices. Static routes are useful for smaller networks with only one path to an outside network or to provide security for certain types of traffic in a larger network.

Configuring Static Routing for IPv6 (CLI)

For configuring static routes for IPv6, see the *Configuring Static Routing for IPv6* section.

For more information about static routes, see the "Implementing Static Routes for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Static Route Support for Object Tracking

The IPv6 Static Route Support for Object Tracking feature allows an IPv6 static route to be associated with a tracked-object. For more information, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Policy-Based Routing for IPv6

Policy-based routing (PBR) gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, which lessens reliance on routes derived from routing protocols. Therefore, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IPv6 precedence. For a simple policy, you can use any one of these tasks; for a complex policy, you can use all of them. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

PBR for IPv6 may be applied to both forwarded and originated IPv6 packets. For forwarded packets, PBR for IPv6 will be implemented as an IPv6 input interface feature, supported in the following forwarding paths:

- Process
- Cisco Express Forwarding (formerly known as CEF)
- Distributed Cisco Express Forwarding

Policies can be based on the IPv6 address, port numbers, protocols, or packet size.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IPv6 precedence bits, giving the network the ability to enable differentiated classes of service.

Note This is not supported on Cisco Catalyst 9500 Series Switches - High Performance

- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific quality of service (QoS) through the network.

PBR allows you to classify and mark packets at the edge of the network. PBR marks a packet by setting precedence value. The precedence value can be used directly by devices in the network core to apply the appropriate QoS to a packet, which keeps packet classification at your network edge.

For enabling PBR for IPv6, see the *Enabling Local PBR for IPv6* section.

For enabling IPv6 PBR for an interface, see the *Enabling IPv6 PBR on an Interface* section.

RIP for IPv6

Routing Information Protocol (RIP) for IPv6 is a distance-vector protocol that uses hop count as a routing metric. It includes support for IPv6 addresses and prefixes and the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages.

For configuring RIP for IPv6, see the *Configuring RIP for IPv6* section.

For more information about RIP for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

OSPF for IPv6

The switch supports Open Shortest Path First (OSPF) for IPv6, a link-state protocol for IP.

For configuring OSPF for IPv6, see the *Configuring OSPF for IPv6* section.

For more information, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IS-IS for IPv6

The switch supports Integrated Intermediate System-to-Intermediate System (IS-IS) for IPv6, an Open Systems Interconnection (OSI) hierarchical routing protocol. For more information, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

EIGRP IPv6

Switches support the Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv6. It is configured on the interfaces on which it runs and does not require a global IPv6 address. Switches running Network Essentials only support EIGRPv6 stub routing.

Before running, an instance of EIGRP IPv6 requires an implicit or explicit router ID. An implicit router ID is derived from a local IPv6 address, so any IPv6 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 IPv6 router ID.

For configuring EIGRP for IPv6, see the *Configuring EIGRP for IPv6* section.

For more information about EIGRP for IPv6, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

EIGRPv6 Stub Routing

The EIGRPv6 stub routing feature, reduces resource utilization by moving routed traffic closer to the end user.

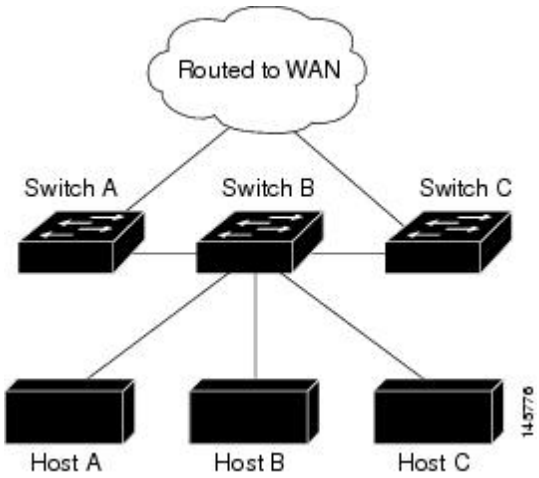
In a network using EIGRPv6 stub routing, the only allowable route for IPv6 traffic to the user is through a switch that is configured with EIGRPv6 stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRPv6 stub routing, you need to configure the distribution and remote routers to use EIGRPv6 and to configure only the switch as a stub. Only specified routes are propagated from the switch. The switch responds to all queries for summaries, connected routes, and routing updates.

Any neighbor that receives a packet informing it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

In the figure given below, switch B is configured as an EIGRPv6 stub router. Switches A and C are connected to the rest of the WAN. Switch B advertises connected, static, redistribution, and summary routes to switch A and C. Switch B does not advertise any routes learned from switch A (and the reverse).

Figure 1: EIGRP Stub Router Configuration



For more information about EIGRPv6 stub routing, see "Implementing EIGRP for IPv6" section of the *Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols, Release 12.4*.

SNMP and Syslog Over IPv6

To support both IPv4 and IPv6, IPv6 network management requires both IPv6 and IPv4 transports. Syslog over IPv6 supports address data types for these transports.

Simple Network Management Protocol (SNMP) and syslog over IPv6 provide these features:

- Support for both IPv4 and IPv6
- IPv6 transport for SNMP and to modify the SNMP agent to support traps for an IPv6 host
- SNMP- and syslog-related MIBs to support IPv6 addressing
- Configuration of IPv6 hosts as trap receivers

For support over IPv6, SNMP modifies the existing IP transport mapping to simultaneously support IPv4 and IPv6. These SNMP actions support IPv6 transport management:

- 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 Manager feature works with IPv6 transport

For information on SNMP over IPv6, including configuration procedures, see the "Managing Cisco IOS Applications over IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

For information about syslog over IPv6, including configuration procedures, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

HTTP(S) Over IPv6

The HTTP client sends requests to both IPv4 and IPv6 HTTP servers, which respond to requests from both IPv4 and IPv6 HTTP clients. URLs with literal IPv6 addresses must be specified in hexadecimal using 16-bit values between colons.

The accept socket call chooses an IPv4 or IPv6 address family. The accept socket is either an IPv4 or IPv6 socket. The listening socket continues to listen for both IPv4 and IPv6 signals that indicate 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.

For more information, see the "Managing Cisco IOS Applications over IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Tool command language (TCL)

Tool command language (TCL) is used in Cisco software for IPv6 to support features such as embedded syslog manager (ESM), embedded event manager (EEM), interactive voice response (IVR), and tclsh parser mode. TCL supports both initiating (client) and listening (server) sockets.

Unsupported IPv6 Unicast Routing Features

The switch does not support these IPv6 features:

- VPN is supported on Cisco Catalyst 9500 Series Switches - High Performance.
- 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 Web Cache Communication Protocol (WCCP)

IPv6 Feature Limitations

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

These are feature limitations.

- The switch cannot forward SNAP-encapsulated IPv6 packets in hardware. They are forwarded in software.
- The switch cannot apply QoS classification on source-routed IPv6 packets in hardware.

IPv6 and Switch Stacks

Note Switch stacks are not supported on Cisco Catalyst 9500 Series Switches - High Performance.

The switch supports IPv6 forwarding across the stack and IPv6 host functionality on the active switch. The active switch runs the IPv6 unicast routing protocols and computes the routing tables. They receive the tables and create hardware IPv6 routes for forwarding. The active switch also runs all IPv6 applications.

If a new switch becomes the active switch, it recomputes the IPv6 routing tables and distributes them to the member switches. While the new active switch is being elected and is resetting, the switch stack does not forward IPv6 packets. The stack MAC address changes, which also changes the IPv6 address. When you specify the stack IPv6 address with an extended unique 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. See the *Configuring IPv6 Addressing and Enabling IPv6 Routing* section.

If you configure the persistent MAC address feature on the stack and the active switch changes, the stack MAC address does not change for approximately 4 minutes.

These are the functions of IPv6 active switch and members:

- Active switch:
  - runs IPv6 routing protocols
  - generates routing tables
  - distributes routing tables to member switches that use distributed Cisco Express Forwarding for IPv6
  - runs IPv6 host functionality and IPv6 applications
- Member switch:
  - receives Cisco Express Forwarding for IPv6 routing tables from the active switch
  - programs the routes into hardware

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

  - flushes the Cisco Express Forwarding for IPv6 tables on active switch re-election

Default IPv6 Configuration

Table 1: Default IPv6 Configuration

Feature	Default Setting
SDM template	Default is core template
IPv6 routing	Disabled globally and on all interfaces

Feature	Default Setting
Cisco Express Forwarding for IPv6 or distributed Cisco Express Forwarding for IPv6	Disabled (IPv4 Cisco Express Forwarding and distributed Cisco Express Forwarding are enabled by default) **Note** When IPv6 routing is enabled, Cisco Express Forwarding for IPv6 and distributed Cisco Express Forwarding for IPv6 are automatically enabled.
IPv6 addresses	None configured

How to Configure IPv6 Unicast Routing

The following sections shows the various configuration options available for IPv6 Unicast Routing

Configuring IPv6 Addressing and Enabling IPv6 Routing

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

Before configuring IPv6 on the switch, consider these guidelines:

- Not all features discussed in this chapter are supported by the switch. See the Unsupported IPv6 Unicast Routing Features.
- In the **ipv6 address** interface configuration command, you must enter the *ipv6-address* and *ipv6-prefix* variables with the address 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).

To forward IPv6 traffic on an interface, you must configure a global IPv6 address on that interface. Configuring an 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

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.

For more information about configuring IPv6 routing, see the “Implementing Addressing and Basic Connectivity for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To assign an IPv6 address to a Layer 3 interface and enable IPv6 routing, perform this procedure:

Procedure

	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	**sdm prefer** {**core** | **distribution** | **nat** | **sda**} **Example:** `Device(config)# sdm prefer core`	Selects an SDM template: • **core**—Sets the switch to the default template. • **distribution**—Sets the distribution template • **nat**—Maximizes the NAT configuration on the switch. • **sda**—Sets the sda template
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**reload** **Example:** `Device# reload`	Reloads the operating system.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode after the switch reloads.
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the Layer 3 interface to configure. The interface can be a physical interface, a switch virtual interface (SVI), or a Layer 3 EtherChannel.
Step 8	**no switchport** **Example:**	Removes the interface from Layer 2 configuration mode (if it is a physical interface).

	Command or Action	Purpose
	`Device(config-if)# no switchport`	

Step 9	Use one of the following:
	• **ipv6 address** *ipv6-prefix/prefix length* **eui-64**
	• **ipv6 address** *ipv6-address/prefix length*
	• **ipv6 address** *ipv6-address* **link-local**
	• **ipv6 enable**
	• **ipv6 address** *WORD*
	• **ipv6 address** *autoconfig*
	• **ipv6 address** *dhcp*
	Example:
	`Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64`
	`Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64`
	`Device(config-if)# ipv6 address 2001:0DB8:c18:1:: link-local`
	`Device(config-if)# ipv6 enable`
		• Specifies a global IPv6 address with an extended unique 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.
		• Manually configures an IPv6 address on the interface.
		• Specifies 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 configures an IPv6 link-local address on the interface, and enables the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.

Step 10	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 11	**ip routing** **Example:** `Device(config)# ip routing`	Enables IP routing on the switch.
Step 12	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 13	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 14	**show ipv6 interface** *interface-id* **Example:** `Device# show ipv6 interface gigabitethernet 1/0/1`	Verifies your entries.
Step 15	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring IPv4 and IPv6 Protocol Stacks

To configure a Layer 3 interface to support both IPv4 and IPv6 and to enable IPv6 routing, perform this procedure:.

Note To disable IPv6 processing on an interface that has not been configured with an IPv6 address, use the **no ipv6 enable** interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ipv6 unicast-routing**
5. **interface** *interface-id*
6. **no switchport**
7. **ip address** *ip-address mask* [**secondary**]
8. Use one of the following:
   - **ipv6 address** *ipv6-prefix/prefix length* **eui-64**
   - **ipv6 address** *ipv6-address/prefix length*
   - **ipv6 address** *ipv6-address* **link-local**
   - **ipv6 enable**
   - **ipv6 address** *WORD*
   - **ipv6 address** *autoconfig*
   - **ipv6 address** *dhcp*
9. **end**
10. Use one of the following:
    - **show interface** *interface-id*
    - **show ip interface** *interface-id*
    - **show ipv6 interface** *interface-id*

11. **copy running-config startup-config**

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	**ip routing** **Example:** `Device(config)# ip routing`	Enables routing on the switch.
Step 4	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables forwarding of IPv6 data packets on the switch.
Step 5	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 6	**no switchport** **Example:** `Device(config-if)# no switchport`	Removes the interface from Layer 2 configuration mode (if it is a physical interface).
Step 7	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.1.2.3 255.255.255`	Specifies a primary or secondary IPv4 address for the interface.
Step 8	Use one of the following: • **ipv6 address** *ipv6-prefix/prefix length* **eui-64** • **ipv6 address** *ipv6-address/prefix length* • **ipv6 address** *ipv6-address* **link-local** • **ipv6 enable**	• Specifies a global IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. • Specifies a link-local address on the interface to be used instead of the automatically configured link-local address when IPv6 is enabled on the interface.

	Command or Action	Purpose
	• **ipv6 address** *WORD* • **ipv6 address** *autoconfig* • **ipv6 address** *dhcp*	• Automatically configures an IPv6 link-local address on the interface, and enables the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link. **Note** To remove all manually configured IPv6 addresses from an interface, use the **no ipv6 address** interface configuration command without arguments.
Step 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 10	Use one of the following: • **show interface** *interface-id* • **show ip interface** *interface-id* • **show ipv6 interface** *interface-id*	Verifies your entries.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Default Router Preference

Router advertisement messages are sent with the default router preference (DRP) configured by the **ipv6 nd router-preference** interface configuration command. If no DRP is configured, RAs are sent with a medium preference.

A DRP is useful when two routers on a link might provide equivalent, but not equal-cost routing, and policy might dictate that hosts should prefer one of the routers.

For more information about configuring DRP for IPv6, see the "Implementing IPv6 Addresses and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure a DRP for a router on an interface, perform this procedure:

Procedure

	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	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode and identifies the Layer 3 interface on which you want to specify the DRP.
Step 4	**ipv6 nd router-preference** {**high** | **medium** | **low**} **Example:** `Device(config-if)# ipv6 nd router-preference medium`	Specifies a DRP for the router on the switch interface.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show ipv6 interface** **Example:** `Device# show ipv6 interface`	Verifies the configuration.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring IPv6 ICMP Rate Limiting

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.

To change the ICMP rate-limiting parameters, perform this procedure:

Procedure

	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** *interval* [*bucketsize*] **Example:** `Device(config)# ipv6 icmp error-interval 50 20`	Configures 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 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 interface** [*interface-id*] **Example:** `Device# show ipv6 interface gigabitethernet0/1`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Cisco Express Forwarding and distributed Cisco Express Forwarding for IPv6

Cisco Express Forwarding is a Layer 3 IP switching technology to improve network performance. Cisco Express Forwarding 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 switch stack, the hardware uses distributed Cisco Express Forwarding in the stack. IPv4 Cisco Express Forwarding and distributed Cisco Express Forwarding are enabled by default. IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding are disabled by default, but automatically enabled when you configure IPv6 routing.

IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding are automatically disabled when IPv6 routing is unconfigured. IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding cannot disabled through configuration. You can verify the IPv6 state by entering the **show ipv6 cef** privileged EXEC command.

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.

For more information about configuring Cisco Express Forwarding and distributed Cisco Express Forwarding, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Configuring Static Routing for IPv6

For more information about configuring static IPv6 routing, see the "Implementing Static Routes for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure static IPv6 routing, perform this procedure:

Before you begin

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.

Procedure

	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** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] **Example:** `Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130`	Configures 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

	Command or Action	Purpose
		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*—Specifies 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 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 5	Use one of the following:
	• **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix length*] [**interface** *interface-id*] [**detail**]][**recursive**] [**detail**]
	• **show ipv6 route static** [*updated*]
	Example:
	`Device# show ipv6 static 2001:0DB8::/32 interface gigabitethernet2/0/1`
	or
	`Device# show ipv6 route static`
		Verifies your entries by displaying the contents of the IPv6 routing table.
		• **interface** *interface-id*—(Optional) Displays only those static routes with the specified interface as an egress interface.
		• **recursive**—(Optional) Displays 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) Displays this additional information:
		• For valid recursive routes, the output path set, and maximum resolution depth.

	Command or Action	Purpose
		• For invalid routes, the reason why the route is not valid.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling IPv6 PBR on an Interface

To enable Policy-Based Routing (PBR) for IPv6, you must create a route map that specifies the packet match criteria and desired policy-route action. Then you associate the route map on the required interface. All packets arriving on the specified interface that match the match clauses will be subject to PBR.

In PBR, the **set vrf** command decouples the virtual routing and forwarding (VRF) instance and interface association and allows the selection of a VRF based on access control list (ACL)-based classification using existing PBR or route-map configurations. It provides a single router with multiple routing tables and the ability to select routes based on ACL classification. The router classifies packets based on ACL, selects a routing table, looks up the destination address, and then routes the packet.

To enable PBR for IPv6, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following:
   - **match length** *minimum-length maximum-length*
   - **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
5. Do one of the following:
   - **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
   - **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
6. **exit**
7. **interface** *type number*
8. **ipv6 policy route-map** *route-map-name*
9. **end**

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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map rip-to-ospf permit`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing, and enters route-map configuration mode.

Step 4	Do one of the following:
	• **match length** *minimum-length maximum-length*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
	Example:
	`Device(config-route-map)# match length 3 200`
	Example:
	`Device(config-route-map)# match ipv6 address marketing`
		Specifies the match criteria.
		• You can specify any or all of the following:
		• Matches the Level 3 length of the packet.
		• Matches a specified IPv6 access list.
		• If you do not specify a **match** command, the route map applies to all packets.

Step 5	Do one of the following:
	• **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	Example:
	`Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
	Example:
	`Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95`
		Specifies the action or actions to take on the packets that match the criteria.
		• You can specify any or all of the following:
		• Sets next hop to which to route the packet (the next hop must be adjacent).
		• Sets next hop to which to route the packet, if there is no explicit route for this destination.

Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 8	**ipv6 policy route-map** *route-map-name* **Example:** `Device(config-if)# ipv6 policy-route-map interactive`	Identifies a route map to use for IPv6 PBR on an interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Local PBR for IPv6

Packets that are generated by the device are not normally policy routed. Perform this task to enable local IPv6 policy-based routing (PBR) for such packets, indicating which route map the device should use.

To enable Local PBR for IPv6, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 local policy route-map** *route-map-name*
4. **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 local policy route-map** *route-map-name* **Example:** `Device(config)# ipv6 local policy route-map pbr-src-90`	Configures IPv6 PBR for packets generated by the device.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring RIP for IPv6

For more information about configuring RIP routing for IPv6, see the "Implementing RIP for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com,

To configure RIP routing for IPv6, perform this procedure:

Before you begin

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.

Procedure

	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 router rip** *name* **Example:** `Device(config)# ipv6 router rip cisco`	Configures an IPv6 RIP routing process, and enters router configuration mode for the process.
Step 4	**maximum-paths** *number-paths* **Example:** `Device(config-router)# maximum-paths 6`	(Optional) Define the maximum number of equal-cost routes that IPv6 RIP can support. The range is from 1 to 32, and the default is 16 routes.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Returns to global configuration mode.
Step 6	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the Layer 3 interface to configure.

	Command or Action	Purpose
Step 7	**ipv6 rip** *name* **enable** **Example:** `Device(config-if)# ipv6 rip cisco enable`	Enables the specified IPv6 RIP routing process on the interface.
Step 8	**ipv6 rip** *name* **default-information** {**only** | **originate**} **Example:** `Device(config-if)# ipv6 rip cisco default-information only`	(Optional) Originates 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 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 10	Use one of the following:
	• **show ipv6 rip** [*name*] [**interface** *interface-id*] [**database**] [**next-hops**]
	• **show ipv6 rip**
	Example:
	`Device# show ipv6 rip cisco interface gigabitethernet 2/0/1`
	or
	`Device# show ipv6 rip`
		• Displays information about current IPv6 RIP processes.
		• Displays the current contents of the IPv6 routing table.

Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring OSPF for IPv6

For more information about configuring OSPF routing for IPv6, see the "Implementing OSPF for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure OSPF routing for IPv6, perform this procedure:

Before you begin

You can customize OSPF for IPv6 for your network. However, the defaults for OSPF in IPv6 are set to meet the requirements of most customers and features.

Follow these guidelines:

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

Procedure

	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 router ospf** *process-id* **Example:** `Device(config)# ipv6 router ospf 21`	Enables 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 4	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*] **Example:** `Device(config)# area .3 range 2001:0DB8::/32 not-advertise`	(Optional) Consolidates and summarizes 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

	Command or Action	Purpose
		address). A slash mark (/) must precede the decimal value. • **advertise**—(Optional) Sets the address range status to advertise and generate a Type 3 summary link-state advertisement (LSA). • **not-advertise**—(Optional) Sets the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and component networks remain hidden from other networks. • **cost** *cost*—(Optional) Sets the 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 5	**maximum paths** *number-paths* **Example:** `Device(config)# maximum paths 16`	(Optional) Defines 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 32, and the default is 16 paths.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 8	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*] **Example:** `Device(config-if)# ipv6 ospf 21 area .3`	Enables OSPF for IPv6 on the interface. • **instance** *instance-id*—(Optional) Instance identifier.
Step 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 10	Use one of the following: • **show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-id*]	• Displays information about OSPF interfaces. • Displays general information about OSPF routing processes.

	Command or Action	Purpose

	• **show ipv6 ospf** [*process-id*] [*area-id*]
	Example:
	`Device# show ipv6 ospf 21 interface gigabitethernet2/0/1`
	or
	`Device# show ipv6 ospf 21`	

Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring EIGRP for IPv6

Before configuring the switch to run IPv6 EIGRP, enable routing by entering the **ip routing global configuration** command, enable the forwarding of IPv6 packets by entering the **ipv6 unicast-routing global** configuration command, and enable IPv6 on any Layer 3 interfaces on which you want to enable IPv6 EIGRP.

To set an explicit router ID, use the **show ipv6 eigrp** command to see the configured router IDs, and then use the **router-id** command.

As with EIGRP IPv4, you can use EIGRPv6 to specify your EIGRP IPv6 interfaces and to select a subset of those as passive interfaces. Use the **passive-interface** command to make an interface passive, and 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.

For more configuration procedures, see the "Implementing EIGRP for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Configuring IPv6 Unicast Reverse Path Forwarding

The unicast Reverse Path Forwarding (unicast RPF) feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

Note

- Do not configure Unicast RPF if the switch is in a mixed hardware stack combining more than one switch type.

For detailed IP unicast RPF configuration information, see the *Other Security Features* chapter in the *Cisco IOS Security Configuration Guide, Release 12.4*.

Configuring DHCP for IPv6 Address Assignment

This section describes only the DHCPv6 address assignment. For more information about configuring the DHCPv6 client, server, or relay agent functions, see the "Implementing DHCP for IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Default DHCPv6 Address Assignment Configuration

By default, no DHCPv6 features are configured on the switch.

DHCPv6 Address Assignment Configuration Guidelines

When configuring DHCPv6 address assignment, consider these guidelines:

- In the procedures, the specified interface must be one of these Layer 3 interfaces:
  - DHCPv6 IPv6 routing must be enabled on a Layer 3 interface.
  - SVI: a VLAN interface created by using the **interface vlan** *vlan_id* command.
  - EtherChannel port channel in Layer 3 mode: a port-channel logical interface created by using the **interface port-channel port-channel-number** command.
- The switch can act as a DHCPv6 client, server, or relay agent. The DHCPv6 client, server, and relay function are mutually exclusive on an interface.
- The DHCPv6 client, server, or relay agent runs only on the active switch. When there is an active switch re-election, the new active switch retains the DHCPv6 configuration. However, the local RAM copy of the DHCP server database lease information is not retained.

Enabling DHCPv6 Server Function (CLI)

Use the **no** form of the DHCP pool configuration mode commands to change the DHCPv6 pool characteristics. To disable the DHCPv6 server function on an interface, use the **no ipv6 dhcp server** interface configuration command.

To enable the DHCPv6 server function on an interface, perform this procedure:

Procedure

	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	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool 7`	Enters DHCP pool configuration mode, and define the name for the IPv6 DHCP pool. The pool name can be a symbolic string (such as Engineering) or an integer (such as 0).
Step 4	**address prefix** *IPv6-prefix* {**lifetime**} {*t1 t1* | **infinite**} **Example:** `Device(config-dhcpv6)# address prefix 2001:1000::0/64 lifetime 3600`	(Optional) Specifies an address prefix for address assignment. This address must be in hexadecimal, using 16-bit values between colons. **lifetime** *t1 t1*—Specifies a time interval (in seconds) that an IPv6 address prefix remains in the valid state. The range is 5 to 4294967295 seconds. Specify **infinite** for no time interval.
Step 5	**link-address** *IPv6-prefix* **Example:** `Device(config-dhcpv6)# link-address 2001:1002::0/64`	(Optional) Specifies a link-address IPv6 prefix. When an address on the incoming interface or a link-address in the packet matches the specified IPv6 prefix, the server uses the configuration information pool. This address must be in hexadecimal, using 16-bit values between colons.
Step 6	**vendor-specific** *vendor-id* **Example:** `Device(config-dhcpv6)# vendor-specific 9`	(Optional) Enters vendor-specific configuration mode and specifies a vendor-specific identification number. This number is the vendor IANA Private Enterprise Number. The range is 1 to 4294967295.
Step 7	**suboption** *number* {**address** *IPv6-address* | **ascii** *ASCII-string* | **hex** *hex-string*} **Example:** `Device(config-dhcpv6-vs)# suboption 1 address 1000:235D::`	(Optional) Enters a vendor-specific suboption number. The range is 1 to 65535. Enter an IPv6 address, ASCII text, or a hex string as defined by the suboption parameters.
Step 8	**exit** **Example:** `Device(config-dhcpv6-vs)# exit`	Returns to DHCP pool configuration mode.
Step 9	**exit** **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-dhcpv6)# exit`	
Step 10	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the interface to configure.
Step 11	**ipv6 dhcp server** [*poolname* | **automatic**] [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server automatic`	Enables DHCPv6 server function on an interface. • *poolname*—(Optional) User-defined name for the IPv6 DHCP pool. The pool name can be a symbolic string (such as Engineering) or an integer (such as 0). • **automatic**—(Optional) Enables the system to automatically determine which pool to use when allocating addresses for a client. • **rapid-commit**—(Optional) Allows two-message exchange method. • **preference** *value*—(Optional) Configures the preference value carried in the preference option in the advertise message sent by the server. The range is from 0 to 255. The preference value default is 0. • **allow-hint**—(Optional) Specifies whether the server should consider client suggestions in the SOLICIT message. By default, the server ignores client hints.
Step 12	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 13	Do one of the following:
	• **show ipv6 dhcp pool**
	• **show ipv6 dhcp interface**
	Example:
	`Device# show ipv6 dhcp pool`
	or
	`Device# show ipv6 dhcp interface`
		• Verifies DHCPv6 pool configuration.
		• Verifies that the DHCPv6 server function is enabled on an interface.

Step 14	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

Enabling DHCPv6 Client Function

To enable the DHCPv6 client on an interface, perform this procedure:

Procedure

	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** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode, and specifies the interface to configure.
Step 4	**ipv6 address dhcp** [**rapid-commit**] **Example:** `Device(config-if)# ipv6 address dhcp rapid-commit`	Enables the interface to acquire an IPv6 address from the DHCPv6 server. **rapid-commit**—(Optional) Allow two-message exchange method for address assignment.
Step 5	**ipv6 dhcp client request** [**vendor-specific**] **Example:** `Device(config-if)# ipv6 dhcp client request vendor-specific`	(Optional) Enables the interface to request the vendor-specific option.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show ipv6 dhcp interface** **Example:** `Device# show ipv6 dhcp interface`	Verifies that the DHCPv6 client is enabled on an interface.

Displaying IPv6

For complete syntax and usage information on these commands, see the Cisco IOS command reference publications.

Table 2: Command for Monitoring IPv6

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

Table 3: Command for Displaying EIGRP IPv6 Information

Command	Purpose
show ipv6 eigrp [*as-number*] *interface*	Displays information about interfaces configured for EIGRP IPv6.

Command	Purpose
show ipv6 eigrp [*as-number*] *neighbor*	Displays the neighbors discovered by EIGRP IPv6.
show ipv6 interface[*as-number*] *traffic*	Displays the number of EIGRP IPv6 packets sent and received.
show ipv6 eigrptopology [*as-number* | *ipv6-address*] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors** | **Base**]	Displays EIGRP entries in the IPv6 topology table.

Configuration Examples for IPv6 Unicast Routing

Configuring IPv6 Addressing and Enabling IPv6 Routing: Example

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.

```
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet0/11
Device(config-if)# no switchport
Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Device(config-if)# end
Device# show ipv6 interface gigabitethernet0/11
GigabitEthernet0/11 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 Default Router Preference: Example

This example shows how to configure a DRP of *high* for the router on an interface.

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ipv6 nd router-preference high
```

```
Device(config-if)# end
```

Configuring IPv4 and IPv6 Protocol Stacks: Example

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

```
Device(config)# ip routing
Device(config)# ipv6 unicast-routing
Device(config)# interface fastethernet1/0/11
Device(config-if)# no switchport
Device(config-if)# ip address 192.168.99.1 255.255.255.0
Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Device(config-if)# end
```

Enabling DHCPv6 Server Function: Example

This example shows how to configure a pool called *engineering* with an IPv6 address prefix:

```
Device# configure terminal
Device(config)# ipv6 dhcp pool engineering
Device(config-dhcpv6)#address prefix 2001:1000::0/64
Device(config-dhcpv6)# end
```

This example shows how to configure a pool called *testgroup* with three link-addresses and an IPv6 address prefix:

```
Device# configure terminal
Device(config)# ipv6 dhcp pool testgroup
Device(config-dhcpv6)# link-address 2001:1001::0/64
Device(config-dhcpv6)# link-address 2001:1002::0/64
Device(config-dhcpv6)# link-address 2001:2000::0/48
Device(config-dhcpv6)# address prefix 2001:1003::0/64
Device(config-dhcpv6)# end
```

This example shows how to configure a pool called *350* with vendor-specific options:

```
Device# configure terminal
Device(config)# ipv6 dhcp pool 350
Device(config-dhcpv6)# address prefix 2001:1005::0/48
Device(config-dhcpv6)# vendor-specific 9
Device(config-dhcpv6-vs)# suboption 1 address 1000:235D::1
Device(config-dhcpv6-vs)# suboption 2 ascii "IP-Phone"
Device(config-dhcpv6-vs)# end
```

Enabling DHCPv6 Client Function: Example

This example shows how to acquire an IPv6 address and to enable the rapid-commit option:

```
Device(config)# interface gigabitethernet2/0/1
```

```
Device(config-if)# ipv6 address dhcp rapid-commit
```

Configuring IPv6 ICMP Rate Limiting: Example

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

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

Configuring Static Routing for IPv6: Example

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

```
Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet 1/0/1 130
```

Example: Enabling PBR on an Interface

In the following example, a route map named pbr-dest-1 is created and configured, specifying packet match criteria and desired policy-route action. PBR is then enabled on GigabitEthernet interface 0/0/1.

```
ipv6 access-list match-dest-1
  permit ipv6 any 2001:DB8:2001:1760::/32
route-map pbr-dest-1 permit 10
  match ipv6 address match-dest-1
  set interface GigabitEthernet 0/0/0
interface GigabitEthernet0/0/1
  ipv6 policy-route-map interactive
```

Example: Enabling Local PBR for IPv6

In the following example, packets with a destination IPv6 address that match the IPv6 address range allowed by access list pbr-src-90 are sent to the device at IPv6 address 2001:DB8:2003:1::95:

```
ipv6 access-list src-90
  permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:DB8:2003:1::95
ipv6 local policy route-map pbr-src-90
```

Configuring RIP for IPv6: Example

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

```
Device(config)# ipv6 router rip cisco
Device(config-router)# maximum-paths 8
```

```
Device(config)# exit
Device(config)# interface gigabitethernet2/0/11
Device(config-if)# ipv6 rip cisco enable
```

Displaying IPv6: Example

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

```
Device# 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>
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

Standards and RFCs

Standard/RFC	Title
RFC 5453	*Reserved IPv6 Interface Identifiers*
RFC 4292	*IP Forwarding Table*
RFC 4293	*Management Information Base for the Internet Protocol (IP)*

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

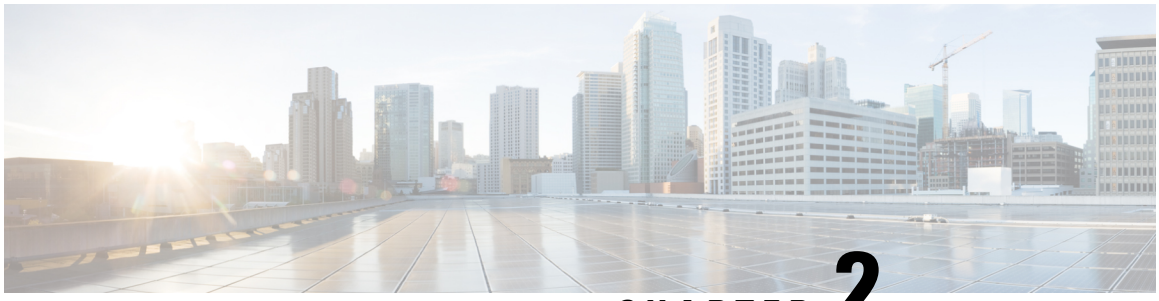
Feature Information

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 4: Feature Information for IPv6 Unicast and Routing

Feature Name	Releases	Feature Information
IPv6 Unicast and Routing	Cisco IOS XE Everest 16.5.1aCisco IOS XE Fuji 16.8.1a	Unicast and routing features supported for IPv6 This feature was introduced for Cisco Catalyst 9500 Series Switches - High Performance



CHAPTER 2

Configuring OSPFv3 Fast Convergence: LSA and SPF Throttling

Understanding Fast Convergence: LSA and SPF Throttling

The Open Shortest Path First version 3 (OSPFv3) link-state advertisement (LSAs) and shortest-path first (SPF) throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability. It also allows faster OSPFv3 convergence by providing LSA rate limiting in milliseconds.

The OSPFv3 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability. It also allows faster OSPFv3 convergence by providing LSA rate limiting in milliseconds.

OSPFv3 can use static timers for rate-limiting SPF calculation and LSA generation. Although these timers are configurable, the values used are specified in seconds, which poses a limitation on OSPFv3 convergence. LSA and SPF throttling achieves subsecond convergence by providing a more sophisticated SPF and LSA rate-limiting mechanism that is able to react quickly to changes and also provide stability and protection during prolonged periods of instability.

How to Configure OSPFv3 Fast Convergence - LSA and SPF Throttling

Tuning LSA and SPF Timers for OSPFv3 Fast Convergence

To tune LSA and SPF Timers for OSPFv3 Fast Convergence, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **timers lsa arrival** *milliseconds*
5. **timers pacing flood** *milliseconds*
6. **timers pacing lsa-group** *seconds*
7. **timers pacing retransmission** *milliseconds*

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	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**timers lsa arrival** *milliseconds* **Example:** `Device(config-router)# timers lsa arrival 300`	Sets the minimum interval at which the software accepts the same LSA from OSPFv3 neighbors.
Step 5	**timers pacing flood** *milliseconds* **Example:** `Device(config-router)# timers pacing flood 30`	Configures LSA flood packet pacing.
Step 6	**timers pacing lsa-group** *seconds* **Example:** `Device(config-router)# timers pacing lsa-group 300`	Changes the interval at which OSPFv3 LSAs are collected into a group and refreshed, checksummed, or aged.
Step 7	**timers pacing retransmission** *milliseconds* **Example:** `Device(config-router)# timers pacing retransmission 100`	Configures LSA retransmission packet pacing in IPv4 OSPFv3.

Configuring LSA and SPF Throttling for OSPFv3 Fast Convergence

To configure LSA and SPF throttling for OSPFv3 fast convergence, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **timers throttle lsa** *start-interval hold-interval max-interval*
6. **timers lsa arrival** *milliseconds*
7. **timers pacing flood** *milliseconds*

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 router ospf** *process-id* **Example:** `Device(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.
Step 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* **Example:** `Device(config-router)# timers throttle spf 200 200 200`	Turns on SPF throttling.
Step 5	**timers throttle lsa** *start-interval hold-interval max-interval* **Example:** `Device(config-router)# timers throttle lsa 300 300 300`	Sets rate-limiting values for OSPFv3 LSA generation.
Step 6	**timers lsa arrival** *milliseconds* **Example:** `Device(config-router)# timers lsa arrival 300`	Sets the minimum interval at which the software accepts the same LSA from OSPFv3 neighbors.

	Command or Action	Purpose
Step 7	**timers pacing flood** *milliseconds* **Example:** `Device(config-router)# timers pacing flood 30`	Configures LSA flood packet pacing.

Configuration Examples for OSPFv3 Fast Convergence - LSA and SPF Throttling

Example: Configuring LSA and SPF Throttling for OSPFv3 Fast Convergence

The following example show how to display the configuration values for SPF and LSA throttling timers:

```
Device# show ipv6 ospf

Routing Process "ospfv3 1" with ID 10.9.4.1
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 It is an autonomous system boundary router
 Redistributing External Routes from,
    ospf 2
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
```

Additional References for Fast Convergence: LSA and SPF Throttling

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

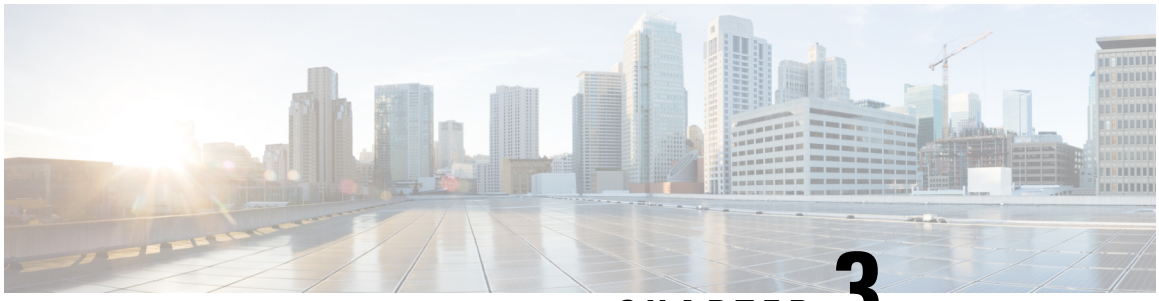
Feature Information for Fast Convergence: LSA and SPF Throttling

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 OSPFv3 Fast Convergence: LSA and SPF Throttling

Feature Name	Releases	Feature Information
OSPFv3 Fast Convergence: LSA and SPF Throttling	Cisco IOS XE Fuji 16.8.1a	The OSPFv3 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability.



CHAPTER 3

Configuring HSRP for IPv6

Prerequisites for Configuring HSRP for IPv6

HSRP version 2 must be enabled on an interface before HSRP for IPv6 can be configured.

Information About Configuring HSRP for IPv6

HSRP for IPv6 Overview

HSRP provides routing redundancy for routing IPv6 traffic not dependent on the availability of any single router. IPv6 hosts learn of available routers through IPv6 neighbor discovery router advertisement messages. These messages are multicast periodically or are solicited by hosts.

An HSRP IPv6 group has a virtual MAC address that is derived from the HSRP group number and a virtual IPv6 link-local address that is, by default, derived from the HSRP virtual MAC address. Periodic messages are sent for the HSRP virtual IPv6 link-local address when the HSRP group is active. These messages stop after a final one is sent when the group leaves the active state.

Note When configuring HSRP for IPv6, you must enable HSRP version 2 (HSRPv2) on the interface.

HSRP IPv6 Virtual MAC Address Range

HSRP IPv6 uses a different virtual MAC address block than does HSRP for IP:

0005.73A0.0000 through 0005.73A0.0FFF (4096 addresses)

HSRP IPv6 UDP Port Number

Port number 2029 has been assigned to HSRP IPv6.

How to Enable HSRP for IPv6

Enabling an HSRP Group for IPv6 Operation

HSRP version 2 must be enabled on an interface before HSRP IPv6 can be configured.

Enabling HSRP Version 2

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby version** {**1** | **2**}

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	**standby version** {**1**	**2**} **Example:** `Device(config-if)# standby version 2`	Changes the version of the HSRP. • Version 1 is the default.

Enabling and Verifying an HSRP Group for IPv6 Operation

In this task, when you enter the **standby ipv6** command, a link-local address is generated from the link-local prefix, and a modified EUI-64 format interface identifier is generated in which the EUI-64 interface identifier is created from the relevant HSRP virtual MAC 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 stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need site-local or globally unique addresses to communicate.

In IPv6, a device on the link advertises in RA messages any site-local and global prefixes, and its willingness to function as a default device for the link. RA messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the 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. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

To enabling and verifying an HSRP group for IPv6, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **standby** [*group-number*] **ipv6** {*link-local-address* | **autoconfig**}
6. **standby** [*group-number*] **preempt** [**delay minimum** *seconds* | **reload** *seconds* | **sync** *seconds*]
7. **standby** [*group-number*] **priority** *priority*
8. **exit**
9. **show standby** [*type number* [*group*]] [**all** | **brief**]
10. **show ipv6 interface** [**brief**] [*interface-type interface-number*] [**prefix**]

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 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams. • The **ipv6 unicast-routing** command must be enabled for HSRP for IPv6 to work.
Step 4	**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 5	**standby** [*group-number*] **ipv6** {*link-local-address* | **autoconfig**} **Example:** `Device(config-if)# standby 1 ipv6 autoconfig`	Activates the HSRP in IPv6.
Step 6	**standby** [*group-number*] **preempt** [**delay minimum** *seconds* | **reload** *seconds* | **sync** *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption and preemption delay.
Step 7	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns the device to privileged EXEC mode.
Step 9	**show standby** [*type number* [*group*]] [**all** | **brief**] **Example:** `Device# show standby`	Displays HSRP information.
Step 10	**show ipv6 interface** [**brief**] [*interface-type interface-number*] [**prefix**] **Example:** `Device# show ipv6 interface GigabitEthernet 0/0/0`	Displays the usability status of interfaces configured for IPv6.

Configuration Examples for HSRP for IPv6

Example: Configuration and Verification for an HSRP Group

The following example shows configuration and verification for an HSRP group for IPv6 that consists of Device1 and Device2. The **show standby** command is issued for each device to verify the device's configuration:

Device 1 configuration

```
interface FastEthernet0/0.100
description DATA VLAN for PCs
encapsulation dot1Q 100
ipv6 address 2001:DB8:CAFE:2100::BAD1:1010/64
standby version 2
standby 101 priority 120
standby 101 preempt delay minimum 30
standby 101 authentication ese
standby 101 track Serial0/1/0.17 90
standby 201 ipv6 autoconfig
standby 201 priority 120
standby 201 preempt delay minimum 30
standby 201 authentication ese
standby 201 track Serial0/1/0.17 90
Device1# show standby
FastEthernet0/0.100 - Group 101 (version 2)
State is Active
2 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.296 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
Active router is local
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Active
2 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.428 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
Active router is local
Standby router is FE80::20F:8FFF:FE37:3B70, priority 100 (expires in 7.856 sec)
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Device 2 configuration

```
interface FastEthernet0/0.100
description DATA VLAN for Computers
encapsulation dot1Q 100
```

```
ipv6 address 2001:DB8:CAFE:2100::BAD1:1020/64
standby version 2
standby 101 preempt
standby 101 authentication ese
standby 201 ipv6 autoconfig
standby 201 preempt
standby 201 authentication ese
Device2# show standby
FastEthernet0/0.100 - Group 101 (version 2)
State is Standby
7 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Standby
7 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
Active router is FE80::212:7FFF:FEC6:8F0C, priority 120 (expires in 7.548 sec)
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Additional References for HSRP for IPv6

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

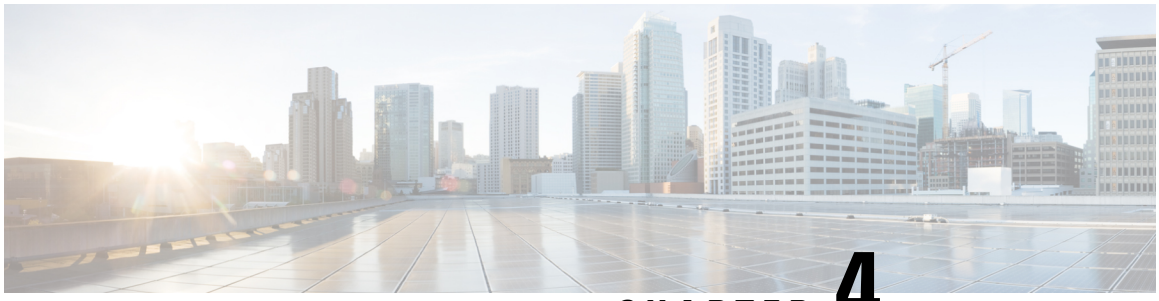
Feature Information for HSRP for IPv6

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 IPv6 over IPv4 GRE Tunnels

Feature Name	Releases	Feature Information
HSRP for IPv6	Cisco IOS XE Fuji 16.8.1a	The HSRP is an FHRP designed to allow for transparent failover of the first-hop IPv6 router.



CHAPTER 4

IPv6 Client IP Address Learning

Prerequisites for IPv6 Client Address Learning

Before configuring IPv6 client address learning, configure the clients to support IPv6.

Information About IPv6 Client Address Learning

Client Address Learning is configured on device to learn the client's IPv4 and IPv6 address and clients transition state maintained by the device on an association, re-association, de-authentication and timeout.

There are three ways for IPv6 client to acquire IPv6 addresses:

- Stateless Address Auto-Configuration (SLACC)
- Stateful DHCPv6
- Static Configuration

For all of these methods, the IPv6 client always sends neighbor solicitation DAD (Duplicate Address Detection) request to ensure there is no duplicate IP address on the network. The device snoops the client's NDP and DHCPv6 packets to learn about its client IP addresses.

SLAAC Address Assignment

The most common method for IPv6 client address assignment is Stateless Address Auto-Configuration (SLAAC). SLAAC provides simple plug-and-play connectivity where clients self-assign an address based on the IPv6 prefix. This process is achieved

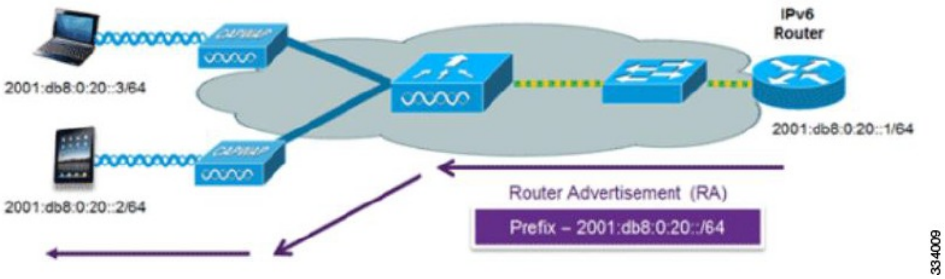
Stateless Address Auto-Configuration (SLAAC) is configured as follows:

- Host sends a router solicitation message.
- Hosts waits for a Router Advertisement message.
- Hosts take the first 64 bits of the IPv6 prefix from the Router Advertisement message and combines it with the 64 bit EUI-64 address (in the case of ethernet, this is created from the MAC Address) to create a global unicast message. The host also uses the source IP address, in the IP header, of the Router Advertisement message, as its default gateway.
- Duplicate Address Detection is performed by IPv6 clients in order to ensure that random addresses that are picked do not collide with other clients.
- The choice of algorithm is up to the client and is often configurable.

The last 64 bits of the IP v6 address can be learned based on the following 2 algorithms:

- EUI-64 which is based on the MAC address of the interface, or
- Private addresses that are randomly generated.

Figure 2: SLAAC Address Assignment

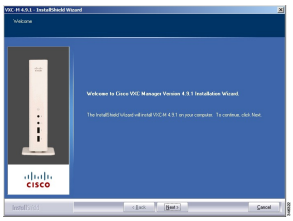


The following Cisco IOS configuration commands from a Cisco-capable IPv6 router are used to enable SLAAC addressing and router advertisements:

```
ipv6 unicast-routing
interface Vlan20
description IPv6-SLAAC
ip address 192.168.20.1 255.255.255.0
ipv6 address FE80:DB8:0:20::1 linklocal
ipv6 address 2001:DB8:0:20::1/64
ipv6 enable
end
```

Stateful DHCPv6 Address Assignment

Figure 3: Stateful DHCPv6 Address Assignment



The use of DHCPv6 is not required for IPv6 client connectivity if SLAAC is already deployed. There are two modes of operation for DHCPv6 called Stateless and Stateful.

The DHCPv6 Stateless mode is used to provide clients with additional network information that is not available in the router advertisement, but not an IPv6 address as this is already provided by SLAAC. This information can include the DNS domain name, DNS server(s), and other DHCP vendor-specific options. This interface configuration is for a Cisco IOS IPv6 router implementing stateless DHCPv6 with SLAAC enabled:

```
ipv6 unicast-routing
ipv6 dhcp pool IPV6_DHCPPOOL
address prefix 2001:db8:5:10::/64
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateless
ip address 192.168.20.1 255.255.255.0
ipv6 nd other-config-flag
ipv6 dhcp server IPV6_DHCPPOOL
ipv6 address 2001:DB8:0:20::1/64
end
```

The DHCPv6 Stateful option, also known as managed mode, operates similarly to DHCPv4 in that it assigns unique addresses to each client instead of the client generating the last 64 bits of the address as in SLAAC. This interface configuration is for a Cisco IOS IPv6 router implementing stateful DHCPv6 on a local Device:

```
ipv6 unicast-routing
ipv6 dhcp pool IPV6_DHCPPOOL
address prefix 2001:db8:5:10::/64
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateful
ip address 192.168.20.1 255.255.255.0
ipv6 address 2001:DB8:0:20::1/64
ipv6 nd prefix 2001:DB8:0:20::/64 no-advertise
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp server IPV6_DHCPPOOL
end
```

This interface configuration is for a Cisco IOS IPv6 router implementing stateful DHCPv6 on an external DHCP server:

```
ipv6 unicast-routing
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateful
ip address 192.168.20.1 255.255.255.0
ipv6 address 2001:DB8:0:20::1/64
```

```
ipv6 nd prefix 2001:DB8:0:20::/64 no-advertise
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp_relay destination 2001:DB8:0:20::2
end
```

Static IP Address Assignment

Statically configured address on a client.

Router Solicitation

A Router Solicitation message is issued by a host to facilitate local routers to transmit Router Advertisement from which it can obtain information about local routing or perform Stateless Auto-configuration. Router Advertisements are transmitted periodically and the host prompts with an immediate Router Advertisement using a Router Solicitation such as - when it boots or following a restart operation.

Router Advertisement

A Router Advertisement message is issued periodically by a router or in response to a Router Solicitation message from a host. The information contained in these messages is used by hosts to perform Stateless Auto-configuration and to modify its routing table.

Neighbor Discovery

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

IPv6 Neighbor Discovery inspection analyzes neighbor discovery messages in order to build a trusted binding table database, and IPv6 neighbor discovery packets that do not comply are dropped. The neighbor binding table in the tracks each IPv6 address and its associated MAC address. Clients are expired from the table according to Neighbor Binding timers.

Neighbor Discovery Suppression

The IPv6 addresses of clients are cached by the device. When the device receives an NS multicast looking for an IPv6 address, and if the target address is known to the device and belongs to one of its clients, the device will reply with an NA message on behalf of the client. The result of this process generates the equivalent of the Address Resolution Protocol (ARP) table of IPv4 but is more efficient - uses generally fewer messages.

Note The device acts like proxy and respond with NA, only when the **ipv6 nd suppress** command is configured

If the device does not have the IPv6 address of a client, the device will not respond with NA and forward the NS packet . To resolve this, an NS Multicast Forwarding knob is provided. If this knob is enabled, the device gets the NS packet for the IPv6 address that it does not have (cache miss) and forwards it . This packet reaches the intended client and the client replies with NA.

This cache miss scenario occurs rarely, and only very few clients which do not implement complete IPv6 stack may not advertise their IPv6 address during NDP.

RA Guard

IPv6 clients configure IPv6 addresses and populate their router tables based on IPv6 router advertisement (RA) packets. The RA guard feature is similar to the RA guard feature of wired networks. RA guard increases the security of the IPv6 network by dropping the unwanted or rogue RA packets that come from clients. If this feature is not configured, malicious IPv6 clients announce themselves as the router for the network often with high priority, which would take higher precedence over legitimate IPv6 routers.

RA-Guard also examines the incoming RA's and decides whether to switch or block them based solely on information found in the message or in the switch configuration. The information available in the frames received is useful for RA validation:

- Port on which the frame is received
- IPv6 source address
- Prefix list

The following configuration information created on the switch is available to RA-Guard to validate against the information found in the received RA frame:

- Trusted/Untrusted ports for receiving RA-guard messages
- Trusted/Untrusted IPv6 source addresses of RA-sender
- Trusted/Untrusted Prefix list and Prefix ranges
- Router Preference

RA guard occurs at the device. You can configure the device to drop RA messages at the device. All IPv6 RA messages are dropped, which protects other clients and upstream wired network from malicious IPv6 clients.

```
//Create a policy for RA Guard//
ipv6 nd raguard policy raguard-router
trusted-port
device-role router

//Applying the RA Guard Policy on port/interface//
interface tengigabitethernet1/0/1 (Katana)
interface gigabitethernet1/0/1 (Edison)

ipv6 nd raguard attach-policy raguard-router
```

Configuring IPv6 Unicast

IPv6 unicasting must always be enabled on the switch . IPv6 unicast routing is disabled.

To configure IPv6 unicast, perform this procedure:

Before you begin

To enable the forwarding of IPv6 unicast datagrams, use the **ipv6 unicast-routing** command in global configuration mode. To disable the forwarding of IPv6 unicast datagrams, use the **no** form of this command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **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.
Step 3	**ipv6 unicast routing** **Example:** `Device(config)# ipv6 unicast routing`	enable the forwarding of IPv6 unicast datagrams

Configuring RA Guard Policy

Configure RA Guard policy on the device to add IPv6 client addresses and populate the router table based on IPv6 router advertisement packets.

To configuring RA guard policy, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd raguard policy `raguard-router`**
4. **trustedport**
5. **device-role router**
6. **exit**

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 nd raguard policy `raguard-router`** **Example:** `Device(config)# ipv6 nd raguard policy raguard-router`	Defines the RA guard policy name and enters RA guard policy configuration mode.
Step 4	**trustedport** **Example:** `Device(config-ra-guard)# trustedport`	(Optional) Specifies that this policy is being applied to trusted ports.
Step 5	**device-role router** **Example:** `Device(config-ra-guard)# device-role router`	Specifies the role of the device attached to the port.
Step 6	**exit** **Example:** `Device(config-ra-guard)# exit`	Exits RA guard policy configuration mode and returns to global configuration mode.

Applying RA Guard Policy

Applying the RA Guard policy on the device will block all the untrusted RA's.

To apply RA guard policy, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tengigabitethernet `1/0/1`**
4. **ipv6 nd raguard attach-policy `raguard-router`**
5. **exit**

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 tengigabitethernet 1/0/1** **Example:** `Device(config)# interface tengigabitethernet 1/0/1`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd raguard attach-policy raguard-router** **Example:** `Device(config-if)# ipv6 nd raguard attach-policy raguard-router`	Applies the IPv6 RA Guard feature to a specified interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring IPv6 Snooping

IPv6 snooping must always be enabled on the switch .

To configuring IPv6 snooping, perform this procedure:

Before you begin

Enable IPv6 on the client machine.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan configuration 1**
4. **ipv6 snooping**
5. **ipv6 nd suppress**
6. **exit**

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	**vlan configuration 1** **Example:** `Device(config)# vlan configuration 1`	Enters VLAN configuration mode.
Step 4	**ipv6 snooping** **Example:** `Device(config-vlan)# ipv6 snooping`	Enables IPv6 snooping on the Vlan.
Step 5	**ipv6 nd suppress** **Example:** `Device(config-vlan-config)# ipv6 nd suppress`	Enables the IPv6 ND suppress on the Vlan.
Step 6	**exit** **Example:** `Device(config-vlan-config)# exit`	Saves the configuration and comes out of the Vlan configuration mode.

Configuring IPv6 ND Suppress Policy

The IPv6 neighbor discovery (ND) multicast suppress feature stops as many ND multicast neighbor solicit (NS) messages as possible by dropping them (and responding to solicitations on behalf of the targets) or converting them into unicast traffic. This feature runs on a layer 2 switch and is used to reduce the amount of control traffic necessary for proper link operations.

When an address is inserted into the binding table, an address resolution request sent to a multicast address is intercepted, and the device either responds on behalf of the address owner or, at layer 2, converts the request into a unicast message and forwards it to its destination.

To configure IPv6 ND suppress policy, perform this procedure:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 nd suppress policy**

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 nd suppress policy** **Example:** `Device(config)# ipv6 nd suppress policy`	Defines the ND suppress policy name and enters ND suppress policy configuration mode.

Configuring IPv6 Snooping on VLAN/PortChannel

Neighbor Discover (ND) suppress can be enabled or disabled on either the VLAN or a switchport.

To configure IPv6 snooping on VLAN/PortChannel, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan config901**
4. **ipv6 nd suppress**
5. **end**
6. **interface gi1/0/1**
7. **ipv6 nd suppress**
8. **end**

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	**vlan config901** **Example:** `Device(config)# vlan config901`	Creates a VLAN and enter the VLAN configuration mode
Step 4	**ipv6 nd suppress** **Example:** `Device(config-vlan)# ipv6 nd suppress`	Applies the IPv6 nd suppress on VLAN.
Step 5	**end** **Example:** `Device(config-vlan)# end`	Exits vlan configuration mode and enters the global configuration mode.
Step 6	**interface gi1/0/1** **Example:** `Device(config)# interface gi1/0/1`	Creates a gigabitethernet port interface.
Step 7	**ipv6 nd suppress** **Example:** `Device(config-vlan)# ipv6 nd suppress`	Applies the IPv6 nd suppress on the interface.
Step 8	**end** **Example:** `Device(config-vlan)# end`	Exits vlan configuration mode and enters the global configuration mode.

Configuring IPv6 on Interface

Follow the procedure given below to configure IPv6 on an interface:

Before you begin

Enable IPv6 on the client and IPv6 support on the wired infrastructure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan 1**
4. **ip address fe80::1 link-local**

5. **ipv6 enable**
6. **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	**interface vlan 1** **Example:** `Device(config)# interface vlan 1`	Creates a interface and enters interface configuration mode.

Step 4	**ip address fe80::1 link-local**
	Example:
	`Device(config-if)# ip address 198.51.100.1 255.255.255.0`
	`Device(config-if)# ipv6 address fe80::1 link-local`
	`Device(config-if)# ipv6 address 2001:DB8:0:1:FFFF:1234::5/64`
	`Device(config-if)# ipv6 address 2001:DB8:0:0:E000::F/64`
		Configures IPv6 address on the interface using the link-local option.

Step 5	**ipv6 enable** **Example:** `Device(config)# ipv6 enable`	(Optional) Enables IPv6 on the interface.
Step 6	**end** **Example:** `Device(config)# end`	Exits from the interface mode.

Configuring DHCP Pool

Follow the procedure given below to configure DHCP Pool on an interface:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 dhcp pool** `Vlan21`
4. **address prefix** `2001:DB8:0:1:FFFF:1234::/64` **lifetime** `300 10`
5. **dns-server** `2001:100:0:1::1`
6. **domain-name** `example.com`
7. **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** `Vlan21` **Example:** `Device(config)# ipv6 dhcp pool vlan1`	Enters the configuration mode and configures the IPv6 DHCP pool on the Vlan.
Step 4	**address prefix** `2001:DB8:0:1:FFFF:1234::/64` **lifetime** `300 10` **Example:** `Device(config-dhcpv6)# address prefix 2001:DB8:0:1:FFFF:1234::/64 lifetime 300 10`	Enters the configuration-dhcp mode and configures the address pool and its lifetime on a Vlan.
Step 5	**dns-server** `2001:100:0:1::1` **Example:** `Device(config-dhcpv6)# dns-server 2001:20:21::1`	Configures the DNS servers for the DHCP pool.
Step 6	**domain-name** `example.com` **Example:** `Device(config-dhcpv6)# domain-name example.com`	Configures the domain name to complete unqualified host names.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring Stateless Auto Address Configuration Without DHCP (CLI)

Follow the procedure given below to configure stateless auto address configuration without DHCP:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan 1**
4. **ip address fe80::1 link-local**
5. **ipv6 enable**
6. **no ipv6 nd managed-config-flag**
7. **no ipv6 nd other-config-flag**
8. **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	**interface vlan 1** **Example:** `Device(config)# interface vlan 1`	Creates a interface and enters interface configuration mode.

Step 4	**ip address fe80::1 link-local**
	Example:
	`Device(config-if)# ip address 198.51.100.1 255.255.255.0`
	`Device(config-if)# ipv6 address fe80::1 link-local`
	`Device(config-if)# ipv6 address 2001:DB8:0:1:FFFF:1234::5/64`
	`Device(config-if)# ipv6 address 2001:DB8:0:0:E000::F/64`
		Configures IPv6 address on the interface using the link-local option.

Step 5	**ipv6 enable** **Example:**	(Optional) Enables IPv6 on the interface.

	Command or Action	Purpose
	`Device(config)# ipv6 enable`	

Step 6	**no ipv6 nd managed-config-flag**
	Example:
	`Device(config)# interface vlan 1`
	`Device(config-if)# no ipv6 nd managed-config-flag`
		Ensures the attached hosts do not use stateful autoconfiguration to obtain addresses.

Step 7	**no ipv6 nd other-config-flag** **Example:** `Device(config-if)# no ipv6 nd other-config-flag`	Ensures the attached hosts do not use stateful autoconfiguration to obtain non-address options from DHCP (domain etc).
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring Stateless Auto Address Configuration With DHCP

Follow the procedure given below to configure stateless auto address configuration with DHCP:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan 1**
4. **ip address fe80::1 link-local**
5. **ipv6 enable**
6. **no ipv6 nd managed-config-flag**
7. **ipv6 nd other-config-flag**
8. **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.

	Command or Action	Purpose
Step 3	**interface vlan 1** **Example:** `Device(config)# interface vlan 1`	Creates a interface and enters interface configuration mode.

Step 4	**ip address fe80::1 link-local**
	Example:
	`Device(config-if)# ip address 198.51.100.1 255.255.255.0`
	`Device(config-if)# ipv6 address fe80::1 link-local`
	`Device(config-if)# ipv6 address 2001:DB8:0:1:FFFF:1234::5/64`
	`Device(config-if)# ipv6 address 2001:DB8:0:0:E000::F/64`
		Configures IPv6 address on the interface using the link-local option.

Step 5	**ipv6 enable** **Example:** `Device(config)# ipv6 enable`	(Optional) Enables IPv6 on the interface.

Step 6	**no ipv6 nd managed-config-flag**
	Example:
	`Device(config)# interface vlan 1`
	`Device(config-if)# no ipv6 nd managed-config-flag`
		Ensures the attached hosts do not use stateful autoconfiguration to obtain addresses.

Step 7	**ipv6 nd other-config-flag** **Example:** `Device(config-if)# no ipv6 nd other-config-flag`	Ensures the attached hosts do not use stateful autoconfiguration to obtain non-address options from DHCP (domain etc).
Step 8	**end** **Example:** `Device(config)# end`	Exits from the interface mode.

Configuring Stateful DHCP Locally

This interface configuration is for a Cisco IOS IPv6 router implementing stateful DHCPv6 on a local Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 dhcp pool IPv6_DHCPPOOL**
5. **address prefix 2001:DB8:0:1:FFFF:1234::/64**
6. **dns-server 2001:100:0:1::1**
7. **domain-name example.com**

8. **exit**
9. **interface vlan1**
10. **description IPv6-DHCP-Stateful**
11. **ipv6 address 2001:DB8:0:20::1/64**
12. **ip address 192.168.20.1 255.255.255.0**
13. **ipv6 nd prefix 2001:db8::/64 no-advertise**
14. **ipv6 nd managed-config-flag**
15. **ipv6 nd other-config-flag**
16. **ipv6 dhcp server IPv6_DHCPPOOL**

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 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Configures IPv6 for unicasting.
Step 4	**ipv6 dhcp pool IPv6_DHCPPOOL** **Example:** `Device (config)# ipv6 dhcp pool IPv6_DHCPPOOL`	Enters the configuration mode and configures the IPv6 DHCP pool on the VLAN.
Step 5	**address prefix 2001:DB8:0:1:FFFF:1234::/64** **Example:** `Device (config-dhcpv6)# address prefix 2001:DB8:0:1:FFFF:1234::/64`	Specifies the address range to provide in the pool.
Step 6	**dns-server 2001:100:0:1::1** **Example:** `Device (config-dhcpv6)# dns-server 2001:100:0:1::1`	Provides the DNS server option to DHCP clients.
Step 7	**domain-name example.com** **Example:** `Device (config-dhcpv6)# domain-name example.com`	Provides the domain name option to DHCP clients.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device (config-dhcpv6)# exit`	Returns to the previous mode.
Step 9	**interface vlan1** **Example:** `Device (config)# interface vlan 1`	Enters the interface mode to configure the stateful DHCP.
Step 10	**description IPv6-DHCP-Stateful** **Example:** `Device (config-if)# description IPv6-DHCP-Stateful`	Enter description for the stateful IPv6 DHCP.
Step 11	**ipv6 address 2001:DB8:0:20::1/64** **Example:** `Device (config-if)# ipv6 address 2001:DB8:0:20::1/64`	Enters the IPv6 address for the stateful IPv6 DHCP.
Step 12	**ip address 192.168.20.1 255.255.255.0** **Example:** `Device (config-if)# ip address 192.168.20.1 255.255.255.0`	Enters the IPv6 address for the stateful IPv6 DHCP.
Step 13	**ipv6 nd prefix 2001:db8::/64 no-advertise** **Example:** `Device (config-if)# ipv6 nd prefix 2001:db8::/64 no-advertise`	Configures the IPv6 routing prefix advertisement that must not be advertised.
Step 14	**ipv6 nd managed-config-flag** **Example:** `Device (config-if)# ipv6 nd managed-config-flag`	Configures IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for address configuration.
Step 15	**ipv6 nd other-config-flag** **Example:** `Device (config-if)# ipv6 nd other-config-flag`	Configures IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for non-address configuration.
Step 16	**ipv6 dhcp server IPv6_DHCPPOOL** **Example:** `Device (config-if)# ipv6 dhcp server IPv6_DHCPPOOL`	Configures the DHCP server on the interface.

Configuring Stateful DHCP Externally

This interface configuration is for a Cisco IOS IPv6 router implementing stateful DHCPv6 on an external DHCP server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **dns-server** `2001:100:0:1::1`
5. **domain-name** `example.com`
6. **exit**
7. **interface** `vlan1`
8. **description IPv6-DHCP-Stateful**
9. **ipv6 address** `2001:DB8:0:20::1/64`
10. **ip address** `192.168.20.1 255.255.255.0`
11. **ipv6 nd prefix** `2001:db8::/64` **no-advertise**
12. **ipv6 nd managed-config-flag**
13. **ipv6 nd other-config-flag**
14. **ipv6 dhcp_relaydestination** `2001:DB8:0:20::2`

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 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Configures the IPv6 for unicasting.
Step 4	**dns-server** `2001:100:0:1::1` **Example:** `Device(config-dhcpv6)# dns-server 2001:100:0:1::1`	Provides the DNS server option to DHCP clients.
Step 5	**domain-name** `example.com` **Example:** `Device(config-dhcpv6)# domain-name example.com`	Provides the domain name option to DHCP clients.
Step 6	**exit** **Example:** `Device(config-dhcpv6)# exit`	Returns to the previous mode.

	Command or Action	Purpose
Step 7	**interface vlan1** **Example:** `Device(config)# interface vlan 1`	Enters the interface mode to configure the stateful DHCP.
Step 8	**description IPv6-DHCP-Stateful** **Example:** `Device(config-if)# description IPv6-DHCP-Stateful`	Enter description for the stateful IPv6 DHCP.
Step 9	**ipv6 address 2001:DB8:0:20::1/64** **Example:** `Device(config-if)# ipv6 address 2001:DB8:0:20::1/64`	Enters the IPv6 address for the stateful IPv6 DHCP.
Step 10	**ip address 192.168.20.1 255.255.255.0** **Example:** `Device(config-if)# ip address 192.168.20.1 255.255.255.0`	Enters the IPv6 address for the stateful IPv6 DHCP.
Step 11	**ipv6 nd prefix 2001:db8::/64 no-advertise** **Example:** `Device(config-if)# ipv6 nd prefix 2001:db8::/64 no-advertise`	Configures the IPv6 routing prefix advertisement that must not be advertised.
Step 12	**ipv6 nd managed-config-flag** **Example:** `Device(config-if)# ipv6 nd managed-config-flag`	Configures IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for address configuration.
Step 13	**ipv6 nd other-config-flag** **Example:** `Device(config-if)# ipv6 nd other-config-flag`	Configures IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for non-address configuration.
Step 14	**ipv6 dhcp_relaydestination 2001:DB8:0:20::2** **Example:** `Device(config-if)# ipv6 dhcp_relay destination 2001:DB8:0:20::2`	Configures the DHCP server on the interface.

Verifying IPv6 Address Learning Configuration

This example displays the output of the **show ipv6 dhcp pool** command. This command displays the IPv6 service configuration on the device. The vlan 21 configured pool detail displays 6 clients that are currently using addresses from the pool.

SUMMARY STEPS

1. **show ipv6 dhcp pool**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ipv6 dhcp pool**
	Example:
	`Deviceshow ipv6 dhcp pool`
	`DHCPv6 pool: vlan21`
	`Address allocation prefix:`
	`2001:DB8:0:1:FFFF:1234::/64 valid 86400 preferred`
	`86400 (6 in use, 0 conflicts)`
	`DNS server: 2001:100:0:1::1`
	`Domain name: example.com`
	`Active clients: 6`
		Displays the IPv6 service configuration on the device.

Additional References

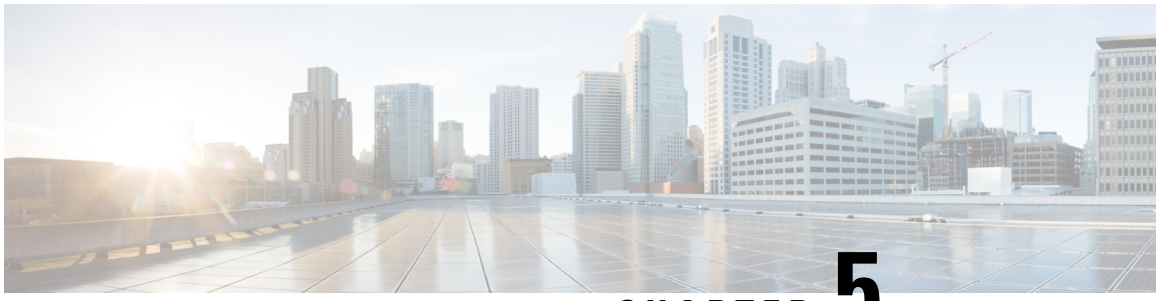
Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

Feature Information for IPv6 Client Address Learning

This table lists the features in this module and provides links to specific configuration information:

Feature	Release	Modification
IPv6 Client Address Learning Functionality	Cisco IOS XE Everest 16.5.1aCisco IOS XE Fuji 16.8.1a	This feature was introduced. This feature was introduced for Cisco Catalyst 9500 Series Switches - High Performance



CHAPTER 5

Implementing IPv6 Multicast

Information About Implementing IPv6 Multicast Routing

This chapter describes how to implement IPv6 multicast routing on the switch.

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IPv6 multicast provides a third scheme, allowing a host to send a single data stream to a subset of all hosts (group transmission) simultaneously.

IPv6 Multicast Overview

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local switch. This signaling is achieved with the MLD protocol.

Switches use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only members of a group can listen to and receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.

IPv6 Multicast Routing Implementation

The Cisco IOS software supports the following protocols to implement IPv6 multicast routing:

- MLD is used by IPv6 switches to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD: MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4, and MLD version 2 is based on version 3 of the IGMP for IPv4. IPv6 multicast for Cisco IOS software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a switch running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

IPv6 Multicast Listener Discovery Protocol

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 switches to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership.

The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

Multicast Queriers and Hosts

A multicast querier is a network device, such as a switch, that sends query messages to discover which network devices are members of a given multicast group.

A multicast host is a receiver, including switches, that send report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the switch alert option set. The switch alert option implies an implementation of the hop-by-hop option header.

MLD Access Group

The MLD access group provides receiver access control in Cisco IOS IPv6 multicast switches. This feature limits the list of groups a receiver can join, and it allows or denies sources used to join SSM channels.

Explicit Tracking of Receivers

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network. This feature also enables the fast leave mechanism to be used with MLD version 2 host reports.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either PIM-SM or PIM-SSM operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM-SM. PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

PIM-SM is used in a multicast network when relatively few switches are involved in each multicast and these switches do not forward multicast packets for a group, unless there is an explicit request for the traffic. PIM-SM distributes information about active sources by forwarding data packets on the shared tree. PIM-SM initially uses shared trees, which requires the use of an RP.

Requests are accomplished via PIM joins, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the RP in the case of a shared tree or the first-hop switch that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups and the hosts that send multicast packets are registered with the RP by that host's first-hop switch.

As a PIM join travels up the tree, switches along the path set up multicast forwarding state so that the requested multicast traffic will be forwarded back down the tree. When multicast traffic is no longer needed, a switch sends a PIM prune up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each switch updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed.

A multicast data sender sends data destined for a multicast group. The designated switch (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then follow the (*, G) multicast tree state in the switches on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

IPv6 BSR: Configure RP Mapping

PIM switches in a domain must be able to map each multicast group to the correct RP address. The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Every PIM-SM multicast group needs to be associated with the IP or IPv6 address of an RP. When a new multicast sender starts sending, its local DR will encapsulate these data packets in a PIM register message and send them to the RP for that multicast group. When a new multicast receiver joins, its local DR will send a PIM join message to the RP for that multicast group. When any PIM switch sends a (*, G) join message, the PIM switch needs to know which is the next switch toward the RP so that G (Group) can send a message to that switch. Also, when a PIM switch is forwarding data packets using (*, G) state, the PIM switch needs to know which is the correct incoming interface for packets destined for G, because it needs to reject any packets that arrive on other interfaces.

A small set of switches from a domain are configured as candidate bootstrap switches (C-BSRs) and a single BSR is selected for that domain. A set of switches within a domain are also configured as candidate RPs (C-RPs); typically, these switches are the same switches that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, and an optional list of group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

Bidirectional BSR support allows bidirectional RPs to be advertised in C-RP messages and bidirectional ranges in the BSM. All switches in a system must be able to use the bidirectional range in the BSM; otherwise, the bidirectional RP feature will not function.

PIM-Source Specific Multicast

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM-SM. However, unlike PIM-SM where data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop switches by MLD membership reports, resulting in shortest-path trees directly to the sources.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IPv6 unicast source address S and the multicast group address G as the IPv6 destination address. Systems will receive this traffic by becoming members of the (S, G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources.

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Before SSM can run with MLD, SSM must be supported in the Cisco IOS IPv6 switch, the host where the application is running, and the application itself.

Routable Address Hello Option

When an IPv6 interior gateway protocol is used to build the unicast routing table, the procedure to detect the upstream switch address assumes the address of a PIM neighbor is always same as the address of the next-hop switch, as long as they refer to the same switch. However, it may not be the case when a switch has multiple addresses on a link.

Two typical situations can lead to this situation for IPv6. The first situation can occur when the unicast routing table is not built by an IPv6 interior gateway protocol such as multicast BGP. The second situation occurs when the address of an RP shares a subnet prefix with downstream switches (note that the RP switch address has to be domain-wide and therefore cannot be a link-local address).

The routable address hello option allows the PIM protocol to avoid such situations by adding a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised. When a PIM switch finds an upstream switch for some address, the result of RPF calculation is compared with the addresses in this option, in addition to the PIM neighbor's address itself. Because this option includes all the possible addresses of a PIM switch on that link, it always includes the RPF calculation result if it refers to the PIM switch supporting this option.

Because of size restrictions on PIM messages and the requirement that a routable address hello option fits within a single PIM hello message, a limit of 16 addresses can be configured on the interface.

PIM IPv6 Stub Routing

The PIM stub routing feature reduces resource usage by moving routed traffic closer to the end user.

In a network using PIM stub routing, the only allowable route for IPv6 traffic to the user is through a switch that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

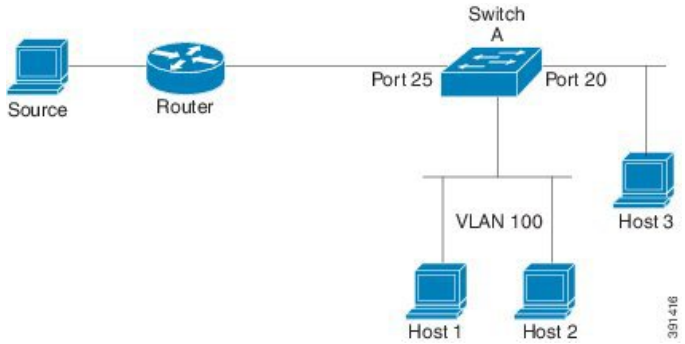
When using PIM stub routing, you should configure the distribution and remote routers to use IPv6 multicast routing and configure only the switch as a PIM stub router. The switch does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the switch. The switch uplink port cannot be used with SVIs.

You must also configure EIGRP stub routing when configuring PIM stub routing on the switch.

The redundant PIM stub router topology is not supported. The redundant topology exists when there is more than one PIM router forwarding multicast traffic to a single access domain. PIM messages are blocked, and the PIM assert and designated router election mechanisms are not supported on the PIM passive interfaces. Only the non-redundant access router topology is supported by the PIM stub feature. By using a non-redundant topology, the PIM passive interface assumes that it is the only interface and designated router on that access domain.

In the figure shown below, Switch A routed uplink port 25 is connected to the router and PIM stub routing is enabled on the VLAN 100 interfaces and on Host 3. This configuration allows the directly connected hosts to receive traffic from multicast source.

Figure 4: PIM Stub Router Configuration



Rendezvous Point

IPv6 PIM provides embedded RP support. Embedded RP support allows the device to learn RP information using the multicast group destination address instead of the statically configured RP. For devices that are the RP, the device must be statically configured as the RP.

The device searches for embedded RP group addresses in MLD reports or PIM messages and data packets. On finding such an address, the device learns the RP for the group from the address itself. It then uses this learned RP for all protocol activity for the group. For devices that are the RP, the device is advertised as an embedded RP must be configured as the RP.

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more devices to operate as an RP. An RP is a single common root placed at a chosen point of a shared distribution tree and is configured statically in each box.

PIM DRs forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways:

- Data is encapsulated in register packets and unicast directly to the RP by the first-hop device operating as the DR.
- If the RP has itself joined the source tree, it is multicast-forwarded per the RPF forwarding algorithm described in the PIM-Sparse Mode section.

The RP address is used by first-hop devices to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop devices to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all devices (including the RP device).

A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for a certain group. The conditions specified by the access list determine for which groups the device is an RP.

IPv6 multicast supports the PIM accept register feature, which is the ability to perform PIM-SM register message filtering at the RP. The user can match an access list or compare the AS path for the registered source with the AS path specified in a route map.

Static Mroutes

IPv6 static mroutes behave much in the same way as IPv4 static mroutes used to influence the RPF check. IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support for RPF checks. Static mroutes support equal-cost multipath mroutes, and they also support unicast-only static routes.

MRIB

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). Its main function is to provide independence between routing protocols and the Multicast Forwarding Information Base (MFIB). It also acts as a coordination and communication point among its clients.

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. Besides routing clients, MRIB also has forwarding clients (MFIB instances)

and special clients such as MLD. MFIB retrieves its forwarding entries from MRIB and notifies the MRIB of any events related to packet reception. These notifications can either be explicitly requested by routing clients or spontaneously generated by the MFIB.

Another important function of the MRIB is to allow for the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also allows for the coordination between MLD and routing protocols.

MFIB

The MFIB is a platform-independent and routing-protocol-independent library for IPv6 software. Its main purpose is to provide a Cisco IOS platform with an interface with which to read the IPv6 multicast forwarding table and notifications when the forwarding table changes. The information provided by the MFIB has clearly defined forwarding semantics and is designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms.

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. The MFIB maintains next-hop address information based on the information in the IPv6 routing table. Because there is a one-to-one correlation between MFIB entries and routing table entries, the MFIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.

MFIB

Note Distributed MFIB has its significance only in a stacked environment where the active switch distributes the MFIB information to the other stack's member switches. In the following section the line cards are nothing but the member switches in the stack.

MFIB (MFIB) is used to switch multicast IPv6 packets on distributed platforms. MFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

MFIB implements the following functions:

- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.

The combination of MFIB and MRIB subsystems also allows the switch to have a "customized" copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

IPv6 Multicast Process Switching and Fast Switching

A unified MFIB is used to provide both fast switching and process switching support for PIM-SM and PIM-SSM in IPv6 multicast. In process switching, the must examine, rewrite, and forward each packet. The packet is first received and copied into the system memory. The switch then looks up the Layer 3 network

address in the routing table. The Layer 2 frame is then rewritten with the next-hop destination address and sent to the outgoing interface. The also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

IPv6 multicast fast switching allows switches to provide better packet forwarding performance than process switching. Information conventionally stored in a route cache is stored in several data structures for IPv6 multicast switching. The data structures provide optimized lookup for efficient packet forwarding.

In IPv6 multicast forwarding, the first packet is fast-switched if the PIM protocol logic allows it. In IPv6 multicast fast switching, the MAC encapsulation header is precomputed. IPv6 multicast fast switching uses the MFIB to make IPv6 destination prefix-based switching decisions. In addition to the MFIB, IPv6 multicast fast switching uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all MFIB entries.

The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created (such as through ARP), a link-layer header for that adjacent node is precomputed and stored in the adjacency table. Once a route is determined, it points to a next hop and corresponding adjacency entry. It is subsequently used for encapsulation during switching of packets.

A route might have several paths to a destination prefix, such as when a switch is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information(NLRI) and next hop (the next switch in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPV6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are only usable for IP unicast, but not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (forexample, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Implementing IPv6 Multicast

Enabling IPv6 Multicast Routing

To enable IPv6 multicast routing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**ipv6 multicast-routing** **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the switch.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Customizing and Verifying the MLD Protocol

Customizing and Verifying MLD on an Interface

To customize and verify MLD on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]} **Example:** `Device(config-if)# ipv6 mld join-group FF04::10`	Configures MLD reporting for a specified group and source.
Step 5	**ipv6 mld access-group** *access-list-name* **Example:** `Device(config-if)# ipv6 access-list acc-grp-1`	Allows the user to perform IPv6 multicast receiver access control.
Step 6	**ipv6 mld static-group** [*group-address*] [**include** | **exclude**] {*source-address* | source-list [*acl*]} **Example:** `Device(config-if)# ipv6 mld static-group ff04::10 include 100::1`	Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.
Step 7	**ipv6 mld query-max-response-time** *seconds* **Example:** `Device(config-if)# ipv6 mld query-timeout 130`	Configures the timeout value before the switch takes over as the querier for the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	**show ipv6 mld groups [link-local]** [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**] **Example:** `Device# show ipv6 mld groups GigabitEthernet 1/0/1`	Displays the multicast groups that are directly connected to the switch and that were learned through MLD.
Step 10	**show ipv6 mld groups summary** **Example:** `Device# show ipv6 mld groups summary`	Displays the number of (*, G) and (S, G) membership reports present in the MLD cache.
Step 11	**show ipv6 mld interface** [*type number*] **Example:** `Device# show ipv6 mld interface GigabitEthernet 1/0/1`	Displays multicast-related information about an interface.

	Command or Action	Purpose
Step 12	**debug ipv6 mld** [*group-name* | *group-address* | *interface-type*] **Example:** `Device# debug ipv6 mld`	Enables debugging on MLD protocol activity.
Step 13	**debug ipv6 mld explicit** [*group-name* | *group-address* **Example:** `Device# debug ipv6 mld explicit`	Displays information related to the explicit tracking of hosts.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implementing MLD Group Limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same switch. The number of MLD limits, globally or per interface, is not configured by default; the limits must be configured by the user. A membership report that exceeds either the per-interface or the global state limit is ignored.

Implementing MLD Group Limits Globally

To implement MLD group limits globally, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld** [*vrf vrf-name*] **state-limit** *number*
4. **copy running-config startup-config**

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 mld** [*vrf vrf-name*] **state-limit** *number* **Example:**	Limits the number of MLD states globally.

	Command or Action	Purpose
	`Device(config)# ipv6 mld state-limit 300`	
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implementing MLD Group Limits per Interface

To implement MLD group limits per interface, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *number*
4. **ipv6 mld limit** *number* [**except**]*access-list*
5. **copy running-config startup-config**

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 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld limit** *number* [**except**]*access-list* **Example:** `Device(config-if)# ipv6 mld limit 100`	Limits the number of MLD states on a per-interface basis.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring Explicit Tracking of Receivers to Track Host Behavior

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

To configuring explicit tracking of receivers to track host behavior, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld explicit-tracking** *access-list-name* **Example:** `Device(config-if)# ipv6 mld explicit-tracking list1`	Enables explicit tracking of hosts.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Resetting the MLD Traffic Counters

To reset the MLD traffic counters, perform this procedure:

Procedure

	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	**clear ipv6 mld traffic** **Example:** `Device# clear ipv6 mld traffic`	Resets all MLD traffic counters.
Step 4	**show ipv6 mld traffic** **Example:**	Displays the MLD traffic counters.

	Command or Action	Purpose
	`Device# show ipv6 mld traffic`	
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clearing the MLD Interface Counters

To clearing the MLD interface counters, perform this procedure

Procedure

	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	**clear ipv6 mld counters** *interface-type* **Example:** `Device# clear ipv6 mld counters Ethernet1/0`	Clears the MLD interface counters.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM

This section explains how to configure PIM.

Configuring PIM-SM and Displaying PIM-SM Information for a Group Range

To configuring PIM-SM and view PIM-SM information for a group range, perform this procedure:

Procedure

	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	ipv6 pim **rp-address** *ipv6-address*[*group-access-list*] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1`	Configures the address of a PIM RP for a particular group range.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 pim interface** [state-on] [**state-off**] [*type-number*] **Example:** `Device# show ipv6 pim interface`	Displays information about interfaces configured for PIM.
Step 6	**show ipv6 pim group-map** [*group-name* | group-address] | [*group-range* | *group-mask*] [**info-source {bsr** | **default** | **embedded-rp** | **static**}] **Example:** `Device# show ipv6 pim group-map`	Displays an IPv6 multicast group mapping table.
Step 7	**show ipv6 pim neighbor** [**detail**] [*interface-type interface-number* | **count**] **Example:** `Device# show ipv6 pim neighbor`	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	**show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*] **Example:** `Device# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim tunnel** [*interface-type interface-number*] **Example:** `Device# show ipv6 pim tunnel`	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**]	Enables debugging on PIM protocol activity.

	Command or Action	Purpose
	Example: `Device# debug ipv6 pim`	
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM Options

To configure PIM options, perform this procedure:

Procedure

	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 pim spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf switch joins the SPT for the specified groups.
Step 4	**ipv6 pim accept-register** {**list** *access-list* | **route-map** *map-name*} **Example:** `Device(config)# ipv6 pim accept-register route-map reg-filter`	Accepts or rejects registers at the RP.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM switch.

	Command or Action	Purpose
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**ipv6 pim join-prune statistic** [*interface-type*] **Example:** `Device(config-if)# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Resetting the PIM Traffic Counters

If PIM malfunctions or in order to verify that the expected number of PIM packets are received and sent, the user can clear PIM traffic counters. Once the traffic counters are cleared, the user can enter the show ipv6 pim traffic command to verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

To resetting the PIM traffic counters, perform this procedure:

Procedure

	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	**clear ipv6 pim traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 4	**show ipv6 pim traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clearing the PIM Topology Table to Reset the MRIB Connection

No configuration is necessary to use the MRIB. However, users may in certain situations want to clear the PIM topology table in order to reset the MRIB connection and verify MRIB information.

To clear the PIM topology table to reset the MRIB connection, perform this procedure:

Procedure

	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	**clear ipv6 pim topology** [*group-name* | *group-address*] **Example:** `Device# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 4	**show ipv6 mrib client** [**filter**] [**name** {*client-name* | *client-name* **: client-id**}] **Example:** `Device# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 5	**show ipv6 mrib route** {**link-local** | **summary** | [*sourceaddress-or-name* | *] [*groupname-or-address*[*prefix-length*]]]	Displays the MRIB route information.

	Command or Action	Purpose
	Example: `Device# show ipv6 mrib route`	
Step 6	**show ipv6 pim topology** [*groupname-or-address* [*sourceaddress-or-name*] | **link-local** | **route-count** [**detail**]] **Example:** `Device# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 7	**debug ipv6 mrib client** **Example:** `Device# debug ipv6 mrib client`	Enables debugging on MRIB client management activity.
Step 8	**debug ipv6 mrib io** **Example:** `Device# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.
Step 9	**debug ipv6 mrib proxy** **Example:** `Device# debug ipv6 mrib proxy`	Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms.
Step 10	**debug ipv6 mrib route** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.
Step 11	**debug ipv6 mrib table** **Example:** `Device# debug ipv6 mrib table`	Enables debugging on MRIB table management activity.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM IPv6 Stub Routing

The PIM Stub routing feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces, uplink PIM interfaces, and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic, it only passes and forwards MLD traffic.

PIM IPv6 Stub Routing Configuration Guidelines

- Before configuring PIM stub routing, you must have IPv6 multicast routing configured on both the stub router and the central router. You must also have PIM mode (sparse-mode) configured on the uplink interface of the stub router.
- The PIM stub router does not route the transit traffic between the distribution routers. Unicast (EIGRP) stub routing enforces this behavior. You must configure unicast stub routing to assist the PIM stub router behavior. For more information, see the *EIGRP Stub Routing* section.
- Only directly connected multicast (MLD) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.
- The redundant PIM stub router topology is not supported.

Default IPv6 PIM Routing Configuration

This table displays the default IPv6 PIM routing configuration for the Device.

Table 7: Default Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

Enabling IPV6 PIM Stub Routing

To enable IPV6 PIM stub routing, perform this procedure:

Before you begin

PIM stub routing is disabled in IPv6 by default.

SUMMARY STEPS

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

3. **ipv6 multicast pim-passive-enable**
4. **interface** *interface-id*
5. **ipv6 pim**
6. **ipv6 pim** {**bsr**} | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | {**passive**}
7. **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 multicast pim-passive-enable** **Example:** `Device(config-if)# ipv6 multicast pim-passive-enable`	Enables IPv6 Multicast PIM routing on the switch.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 9/0/6`	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. You will also need to enable IP PIM sparse mode on the interface, and join the interface as a statically connected member to an MLD static group. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. You will also need to enable IP PIM sparse mode on the VLAN, join the VLAN as a statically connected member to an MLD static group, and then enable MLD snooping on the VLAN, the MLD static group, and physical interface. These interfaces must have IPv6 addresses assigned to them.

	Command or Action	Purpose
Step 5	**ipv6 pim** **Example:** `Device(config-if)# ipv6 pim`	Enables the PIM on the interface.
Step 6	**ipv6 pim** {**bsr**} | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | {**passive**} **Example:** `Device(config-if)# ipv6 pim bsr|dr-priority|hello-interval|join-prune-interval|passive`	Configures the various PIM stub features on the interface. Enter **bsr** to configure BSR on a PIM switch Enter **dr-priority** to configure the DR priority on a PIM switch. Enter **hello-interval** to configure the frequency of PIM hello messages on an interface. Enter **join-prune-interval** to configure periodic join and prune announcement intervals for a specified interface. Enter **passive** to configure the PIM in the passive mode.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Monitoring IPv6 PIM Stub Routing

Table 8: PIM Stub Configuration show Commands

Command	Purpose
show ipv6 pim interface `Device# show ipv6 pim interface`	Displays the PIM stub that is enabled on each interface.
show ipv6 mld groups `Device# show ipv6 mld groups`	Displays the interested clients that have joined the specific multicast source group.
show ipv6 mroute `Device# show ipv6 mroute`	Verifies that the multicast stream forwards from the source to the interested clients.

Disabling Embedded RP Support in IPv6 PIM

A user might want to disable embedded RP support on an interface if all of the devices in the domain do not support embedded RP.

Note This task disables PIM completely, not just embedded RP support in IPv6 PIM.

To disabling embedded RP support in IPv6 PIM, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **enable**
3. **configure terminal**
4. **no ipv6 pim** [**vrf** *vrf-name*] **rp embedded**
5. **interface** *type number*
6. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**no ipv6 pim** [**vrf** *vrf-name*] **rp embedded** **Example:** `Device(config)# no ipv6 pim rp embedded`	Disables embedded RP support in IPv6 PIM.
Step 5	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**no ipv6 pim** **Example:** `Device(config-if)# no ipv6 pim`	Turns off IPv6 PIM on a specified interface.

Configuring a BSR

The tasks included here are described below.

Configuring a BSR and Verifying BSR Information

To configure and verify BSR Information, perform this procedure:

Procedure

	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 pim bsr candidate bsr** *ipv6-address*[*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10`	Configures a switch to be a candidate BSR.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 7	**show ipv6 pim** bsr {**election** | **rp-cache** | **candidate-rp**} **Example:** `Device(config-if)# show ipv6 pim bsr election`	Displays information related to PIM BSR protocol processing.

	Command or Action	Purpose
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Sending PIM RP Advertisements to the BSR

To sending PIM RP advertisements to the BSR, perform this procedure:

Procedure

	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 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] **[priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0`	Sends PIM RP advertisements to the BSR.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring BSR for Use Within Scoped Zones

To configure BSR for use within scoped zones, perform this procedure:

Procedure

	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 pim bsr candidate rp** *ipv6-address* [*hash-mask-length*] [**priority** priority-value] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4`	Configures a switch to be a candidate BSR.
Step 4	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6`	Configures the candidate RP to send PIM RP advertisements to the BSR.
Step 5	**interface** *type number* **Example:** `Device(config-if)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 multicast boundary scope** *scope-value* **Example:** `Device(config-if)# ipv6 multicast boundary scope 6`	Configures a multicast boundary on the interface for a specified scope.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring BSR Switches to Announce Scope-to-RP Mappings

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. A user might want to configure a BSR switch to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. Enabling this feature also allows an RP positioned outside the enterprise's BSR domain to be learned by the known remote RP on the local candidate BSR switch.

To configure BSR switches to announce Scope-to-RP mappings, perform this procedure:

Procedure

	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 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0`	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring SSM Mapping

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the switch will look up the source of a multicast MLD version 1 report from a DNS server.

You can use either DNS-based or static SSM mapping, depending on your switch configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.

Note To use DNS-based SSM mapping, the switch needs to find at least one correctly configured DNS server, to which the switch may be directly attached.

To configuring SSM mapping, perform this procedure:

Procedure

	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	**ipv6 mld ssm-map enable** **Example:** `Device(config)# ipv6 mld ssm-map enable`	Enables the SSM mapping feature for groups in the configured SSM range.
Step 4	**no ipv6 mld ssm-map query dns** **Example:** `Device(config)# no ipv6 mld ssm-map query dns`	Disables DNS-based SSM mapping.
Step 5	**ipv6 mld ssm-map static** *access-list source-address* **Example:** `Device(config-if)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1`	Configures static SSM mappings.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 7	**show ipv6 mld ssm-map** [*source-address*] **Example:** `Device(config-if)# show ipv6 mld ssm-map`	Displays SSM mapping information.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring Static Mroutes

Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your switch to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

To configure static mroutes, perform this procedure:

Procedure

	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 route** {*ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | *unicast* | *multicast*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8::/64 6::6 100`	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 mroute** [link-local | [*group-name* | *group-address* [*source-address* | *source-name*]] **[summary]** [**count**] **Example:** `Device# show ipv6 mroute ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 6	**show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*] **Example:** `Device(config-if)# show ipv6 mroute active`	Displays the active multicast streams on the switch.
Step 7	**show ipv6 rpf** [*ipv6-prefix*] **Example:** `Device(config-if)#  show ipv6 rpf 2001::1:1:2`	Checks RPF information for a given unicast host address and prefix.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Using MFIB in IPv6 Multicast

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled.

Verifying MFIB Operation in IPv6 Multicast

To verify MFIB operation in IPv6 multicast

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show ipv6 mfib** [**linkscope** | **verbose** | *group-address-name* | *ipv6-prefix / prefix-length* | *source-address-name* | **count** | **interface** | **status** | **summary**] **Example:** `Device# show ipv6 mfib`	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	**show ipv6 mfib** [all | linkscope | group-name | group-address [source-name | source-address]] **count** **Example:** `Device# show ipv6 mfib ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 4	**show ipv6 mfib interface** **Example:** `Device# show ipv6 mfib interface`	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 5	**show ipv6 mfib status** **Example:** `Device# show ipv6 mfib status`	Displays general MFIB configuration and operational status.
Step 6	**show ipv6 mfib summary** **Example:** `Device# show ipv6 mfib summary`	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 7	**debug ipv6 mfib** [*group-name* | *group-address*] [**adjacency** | **db** | **fs** | **init** | **interface** | **mrib** [**detail**] | **nat** | **pak** | **platform** | **ppr** | **ps** | **signal** | **table**] **Example:** `Device# debug ipv6 mfib FF04::10 pak`	Enables debugging output on the IPv6 MFIB.

Resetting MFIB Traffic Counters

To reset MFIB traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**clear ipv6 mfib counters** [*group-name* | **group-address** [*source-address* | *source-name*]] **Example:** `Device# clear ipv6 mfib counters FF04::10`	Resets all active MFIB traffic counters.

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

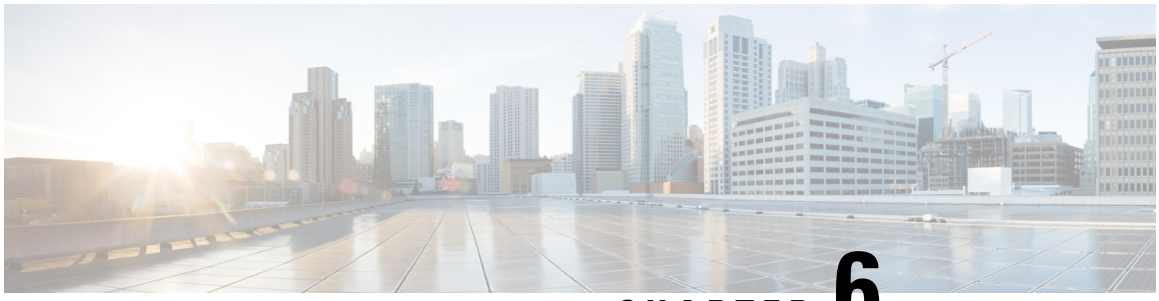
Feature Information

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 9: Feature Information for IPv6 Multicast

Feature Name	Releases	Feature Information
IPv6 multicast	Cisco IOS XE Everest 16.5.1aCisco IOS XE Fuji 16.8.1a	Multicast features for IPv6 This feature was introduced for Cisco Catalyst 9500 Series Switches - High Performance



CHAPTER 6

Configuring Multiprotocol BGP Extensions for IPv6

Information About Configuring Multiprotocol BGP Extensions for IPv6

Understanding Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP is the supported Exterior Gateway Protocol (EGP) for IPv6. Multiprotocol BGP extensions for IPv6 supports many of the same features and functionality as IPv4 BGP. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and Network Layer Reachability Information (NLRI) and next hop (the next device in the path to the destination) attributes that use IPv6 addresses.

How to Implement Multiprotocol BGP for IPv6

Configuring an IPv6 BGP Routing Process and BGP Router ID

Perform this task to configure an IPv6 BGP routing process and an optional BGP router ID for a BGP-speaking device.

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is 32-bit value that is often represented by an IPv4 address. By default, the router ID is set to the IPv4 address of a loopback interface on the device. If no loopback interface is configured on the device, then the software chooses the highest IPv4 address configured to a physical interface on the device to represent the BGP router ID.

When configuring BGP on a device that is enabled only for IPv6 (that is, the device does not have an IPv4 address), you must manually configure the BGP router ID for the device. The BGP router ID, which is represented as a 32-bit value using an IPv4 address syntax, must be unique to the BGP peers of the device.

To configure an IPv6 BGP routing process and BGP router ID, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **bgp router-id** *ip-address*

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process, and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** command unless you configure the **no bgp default ipv4-unicast** command before configuring the **neighbor remote-as** command.
Step 5	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 192.168.99.70`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.

Configuring IPv6 Multiprotocol BGP Between Two Peers

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

To configuring IPv6 multiprotocol BGP between two peers, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
5. **address-family ipv6** [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** `Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	**address-family ipv6** [**unicast** | **multicast**] **Example:**	Specifies the IPv6 address family and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-router)# address-family ipv6`	• The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.

Advertising IPv4 Routes Between IPv6 BGP Peers

If an IPv6 network is connecting two separate IPv4 networks, IPv6 can be used to advertise the IPv4 routes. Configure the peering using the IPv6 addresses within the IPv4 address family. Set the next hop with a static route or with an inbound route map because the advertised next hop will usually be unreachable. Advertising IPv6 routes between two IPv4 peers is also possible using the same model.

To advertising IPv4 routes between IPv6 BGP peers, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**}
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address* [*... ip-address*] [**peer-address**]

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	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor 6peers peer-group`	Creates a multiprotocol BGP peer group.
Step 5	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** `Device(config-router)# neighbor 6peers remote-as 65002`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**neighbor** *ipv6-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 2001:DB8:1234::2 peer-group 6peers`	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 6peers route-map rmap out`	Applies a route map to incoming or outgoing routes. • Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the **clear bgp ipv6** command with the **soft** and **in** keywords will perform a soft reset.
Step 9	**exit** **Example:**	Exits address family configuration mode, and returns the device to router configuration mode.

	Command or Action	Purpose
	`Device(config-router-af)# exit`	
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode, and returns the device to global configuration mode.
Step 11	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map rmap permit 10`	Defines a route map and enters route-map configuration mode.
Step 12	**set ip next-hop** *ip-address* [*... ip-address*] [**peer-address**] **Example:** `Device(config-route-map)# set ip next-hop 10.21.8.10`	Overrides the next hop advertised to the peer for IPv4 packets.

Clearing External BGP Peers

To clear external BGP peers, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**]
3. **clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**] **Example:** `Device# clear bgp ipv6 unicast external soft in`	Clears external IPv6 BGP peers.
Step 3	**clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name* **Example:** `Device# clear bgp ipv6 unicast peer-group marketing`	Clears all members of an IPv6 BGP peer group.

Configuration Examples for Multiprotocol BGP for IPv6

Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer

The following example enables IPv6 globally, configures a BGP process, and establishes a BGP router ID. Also, the IPv6 multiprotocol BGP peer 2001:DB8:0:CC00::1 is configured and activated.

```
ipv6 unicast-routing
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp router-id 192.168.99.70
 neighbor 2001:DB8:0:CC00::1 remote-as 64600
 address-family ipv6 unicast
  neighbor 2001:DB8:0:CC00::1 activate
```

Example: Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 2001:DB8:0:CC00::1 remote-as 64600
address-family ipv6 unicast
 neighbor group1 activate
 neighbor 2001:DB8:0:CC00::1 peer-group group1
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
router bgp 65000
 no bgp default ipv4-unicast
address-family ipv6 unicast
  network 2001:DB8::/24
```

Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

The following example configures the route map named rtp to permit IPv6 unicast routes from network 2001:DB8::/24 if they match the prefix list named cisco:

```
router bgp 64900
no bgp default ipv4-unicast
neighbor 2001:DB8:0:CC00::1 remote-as 64700
address-family ipv6 unicast
 neighbor 2001:DB8:0:CC00::1 activate
```

```
 neighbor 2001:DB8:0:CC00::1 route-map rtp in
ipv6 prefix-list cisco seq 10 permit 2001:DB8::/24
route-map rtp permit 10
 match ipv6 address prefix-list cisco
```

Example: Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes RIP routes into the IPv6 unicast database of the local device:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 unicast
 redistribute rip
```

Example: Advertising IPv4 Routes Between IPv6 Peers

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. Peering is configured using IPv6 addresses in the IPv4 address family configuration mode. The inbound route map named rmap sets the next hop because the advertised next hop is likely to be unreachable.

```
router bgp 65000
!
 neighbor 6peers peer-group
 neighbor 2001:DB8:1234::2 remote-as 65002
 address-family ipv4
 neighbor 6peers activate
 neighbor 6peers soft-reconfiguration inbound
 neighbor 2001:DB8:1234::2 peer-group 6peers
 neighbor 2001:DB8:1234::2 route-map rmap in
!
route-map rmap permit 10
 set ip next-hop 10.21.8.10
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

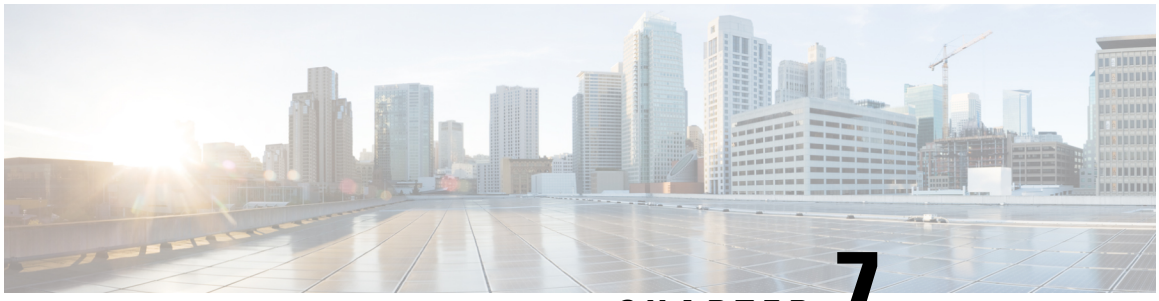
Feature Information

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 10: Feature Information for IPv6 Routing Multiprotocol BGP Extensions for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: Multiprotocol BGP Extensions for IPv6	Cisco IOS XE Fuji 16.8.1a	Multiprotocol BGP extensions for IPv6 supports the same features and functionality as IPv4 BGP.



CHAPTER 7

Configuring MLD Snooping

This module contains details of configuring MLD snooping

Information About Configuring IPv6 MLD Snooping

You can use Multicast Listener Discovery (MLD) snooping to enable efficient distribution of IP Version 6 (IPv6) multicast data to clients and routers in a switched network on the switch. Unless otherwise noted, the term switch refers to a standalone switch and to a switch stack.

To use IPv6, you must configure the IPv6 Switch Database Management (SDM) template on the switch.

For complete syntax and usage information for the commands used in this chapter, see the *Command Reference (Catalyst 9500 Series Switches)* .

Understanding MLD Snooping

In IP Version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on the links that are directly attached to the routers and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD Version 1 (MLDv1) is equivalent to IGMPv2, and MLD Version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol Version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

The switch supports two versions of MLD snooping:

- MLDv1 snooping detects MLDv1 control packets and sets up traffic bridging based on IPv6 destination multicast addresses.
- MLDv2 basic snooping (MBSS) uses MLDv2 control packets to set up traffic forwarding based on IPv6 destination multicast addresses.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 multicast addresses.

Note The switch does not support MLDv2 enhanced snooping, which sets up IPv6 source and destination multicast address-based forwarding.

MLD snooping can be enabled or disabled globally or per VLAN. When MLD snooping is enabled, a per-VLAN IPv6 multicast address table is constructed in software and hardware. The switch then performs IPv6 multicast-address based bridging in hardware.

According to IPv6 multicast standards, the switch derives the MAC multicast address by performing a logical-OR of the four low-order octets of the switch MAC address with the MAC address of 33:33:00:00:00:00. For example, the IPv6 MAC address of FF02:DEAD:BEEF:1:3 maps to the Ethernet MAC address of 33:33:00:01:00:03.

A multicast packet is unmatched when the destination IPv6 address does not match the destination MAC address. The switch forwards the unmatched packet in hardware based the MAC address table. If the destination MAC address is not in the MAC address table, the switch floods the packet to all ports in the same VLAN as the receiving port.

MLD Messages

MLDv1 supports three types of messages:

- Listener Queries are the equivalent of IGMPv2 queries and are either General Queries or Multicast-Address-Specific Queries (MASQs).
- Multicast Listener Reports are the equivalent of IGMPv2 reports
- Multicast Listener Done messages are the equivalent of IGMPv2 leave messages.

MLDv2 supports MLDv2 queries and reports, as well as MLDv1 Report and Done messages.

Message timers and state transitions resulting from messages being sent or received are the same as those of IGMPv2 messages. MLD messages that do not have valid link-local IPv6 source addresses are ignored by MLD routers and switches.

MLD Queries

The switch sends out MLD queries, constructs an IPv6 multicast address database, and generates MLD group-specific and MLD group-and-source-specific queries in response to MLD Done messages. The switch also supports report suppression, report proxying, Immediate-Leave functionality, and static IPv6 multicast group address configuration.

When MLD snooping is disabled, all MLD queries are flooded in the ingress VLAN.

When MLD snooping is enabled, received MLD queries are flooded in the ingress VLAN, and a copy of the query is sent to the CPU for processing. From the received query, MLD snooping builds the IPv6 multicast

address database. It detects multicast router ports, maintains timers, sets report response time, learns the querier IP source address for the VLAN, learns the querier port in the VLAN, and maintains multicast-address aging.

Note When the IPv6 multicast router is a Catalyst 6500 switch and you are using extended VLANs (in the range 1006 to 4094), IPv6 MLD snooping must be enabled on the extended VLAN on the Catalyst 6500 switch in order for the Catalyst 2960, 2960-S, 2960-C, 2960-X or 2960-CX switch to receive queries on the VLAN. For normal-range VLANs (1 to 1005), it is not necessary to enable IPv6 MLD snooping on the VLAN on the Catalyst 6500 switch.

When a group exists in the MLD snooping database, the switch responds to a group-specific query by sending an MLDv1 report. When the group is unknown, the group-specific query is flooded to the ingress VLAN.

When a host wants to leave a multicast group, it can send out an MLD Done message (equivalent to IGMP Leave message). When the switch receives an MLDv1 Done message, if Immediate- Leave is not enabled, the switch sends an MASQ to the port from which the message was received to determine if other devices connected to the port should remain in the multicast group.

Multicast Client Aging Robustness

You can configure port membership removal from addresses based on the number of queries. A port is removed from membership to an address only when there are no reports to the address on the port for the configured number of queries. The default number is 2.

Multicast Router Discovery

Like IGMP snooping, MLD snooping performs multicast router discovery, with these characteristics:

- Ports configured by a user never age out.
- Dynamic port learning results from MLDv1 snooping queries and IPv6 PIMv2 packets.
- If there are multiple routers on the same Layer 2 interface, MLD snooping tracks a single multicast router on the port (the router that most recently sent a router control packet).
- Dynamic multicast router port aging is based on a default timer of 5 minutes; the multicast router is deleted from the router port list if no control packet is received on the port for 5 minutes.
- IPv6 multicast router discovery only takes place when MLD snooping is enabled on the switch.
- Received IPv6 multicast router control packets are always flooded to the ingress VLAN, whether or not MLD snooping is enabled on the switch.
- After the discovery of the first IPv6 multicast router port, unknown IPv6 multicast data is forwarded only to the discovered router ports (before that time, all IPv6 multicast data is flooded to the ingress VLAN).

MLD Reports

The processing of MLDv1 join messages is essentially the same as with IGMPv2. When no IPv6 multicast routers are detected in a VLAN, reports are not processed or forwarded from the switch. When IPv6 multicast routers are detected and an MLDv1 report is received, an IPv6 multicast group address is entered in the VLAN MLD database. Then all IPv6 multicast traffic to the group within the VLAN is forwarded using this address. When MLD snooping is disabled, reports are flooded in the ingress VLAN.

When MLD snooping is enabled, MLD report suppression, called listener message suppression, is automatically enabled. With report suppression, the switch forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers. When MLD snooping is disabled, report suppression is disabled, and all MLDv1 reports are flooded to the ingress VLAN.

The switch also supports MLDv1 proxy reporting. When an MLDv1 MASQ is received, the switch responds with MLDv1 reports for the address on which the query arrived if the group exists in the switch on another port and if the port on which the query arrived is not the last member port for the address.

MLD Done Messages and Immediate-Leave

When the Immediate-Leave feature is enabled and a host sends an MLDv1 Done message (equivalent to an IGMP leave message), the port on which the Done message was received is immediately deleted from the group.You enable Immediate-Leave on VLANs and (as with IGMP snooping), you should only use the feature on VLANs where a single host is connected to the port. If the port was the last member of a group, the group is also deleted, and the leave information is forwarded to the detected IPv6 multicast routers.

When Immediate Leave is not enabled in a VLAN (which would be the case when there are multiple clients for a group on the same port) and a Done message is received on a port, an MASQ is generated on that port. The user can control when a port membership is removed for an existing address in terms of the number of MASQs. A port is removed from membership to an address when there are no MLDv1 reports to the address on the port for the configured number of queries.

The number of MASQs generated is configured by using the **ipv6 mld snooping last-listener-query count** global configuration command. The default number is 2.

The MASQ is sent to the IPv6 multicast address for which the Done message was sent. If there are no reports sent to the IPv6 multicast address specified in the MASQ during the switch maximum response time, the port on which the MASQ was sent is deleted from the IPv6 multicast address database. The maximum response time is the time configured by using the **ipv6 mld snooping last-listener-query-interval** global configuration command. If the deleted port is the last member of the multicast address, the multicast address is also deleted, and the switch sends the address leave information to all detected multicast routers.

When Immediate Leave is not enabled and a port receives an MLD Done message, the switch generates MASQs on the port and sends them to the IPv6 multicast address for which the Done message was sent. You can optionally configure the number of MASQs that are sent and the length of time the switch waits for a response before deleting the port from the multicast group.

When you enable MLDv1 Immediate Leave, the switch immediately removes a port from a multicast group when it detects an MLD Done message on that port. You should only use the Immediate-Leave feature when there is a single receiver present on every port in the VLAN. When there are multiple clients for a multicast group on the same port, you should not enable Immediate-Leave in a VLAN.

Topology Change Notification Processing

When topology change notification (TCN) solicitation is enabled by using the **ipv6 mld snooping tcn query solicit** global configuration command, MLDv1 snooping sets the VLAN to flood all IPv6 multicast traffic with a configured number of MLDv1 queries before it begins sending multicast data only to selected ports. You set this value by using the **ipv6 mld snooping tcn flood query count** global configuration command. The default is to send two queries. The switch also generates MLDv1 global Done messages with valid link-local IPv6 source addresses when the switch becomes the STP root in the VLAN or when it is configured by the user. This is same as done in IGMP snooping.

How to Configure IPv6 MLD Snooping

Default MLD Snooping Configuration

Table 11: Default MLD Snooping Configuration

Feature	Default Setting
MLD snooping (Global)	Disabled.
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query count	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query interval	Global: 1000 (1 second); VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2.
MLD listener suppression	Disabled.

MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch.

- The maximum number of address entries allowed for the switch or switch stack is 4000.

Enabling or Disabling MLD Snooping on the Switch

By default, IPv6 MLD snooping is globally disabled on the switch and enabled on all VLANs. When MLD snooping is globally disabled, it is also disabled on all VLANs. When you globally enable MLD snooping, the VLAN configuration overrides the global configuration. That is, MLD snooping is enabled only on VLAN interfaces in the default state (enabled).

You can enable and disable MLD snooping on a per-VLAN basis or for a range of VLANs, but if you globally disable MLD snooping, it is disabled in all VLANs. If global snooping is enabled, you can enable or disable VLAN snooping.

To globally enable MLD snooping on the switch, perform this procedure:

Procedure

	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 mld snooping** **Example:** `Device(config)# ipv6 mld snooping`	Enables MLD snooping on the switch.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device(config)# copy running-config startup-config`	(Optional) Save your entries in the configuration file.
Step 6	**reload** **Example:**	Reload the operating system.

	Command or Action	Purpose
	`Device(config)# reload`	

Enabling or Disabling MLD Snooping on a VLAN

To enable MLD snooping on a VLAN, perform this procedure:

Procedure

	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 mld snooping** **Example:** `Device(config)# ipv6 mld snooping`	Enables MLD snooping on the switch.
Step 4	**ipv6 mld snooping vlan** *vlan-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1`	Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. **Note** MLD snooping must be globally enabled for VLAN snooping to be enabled.
Step 5	**end** **Example:** `Device(config)# ipv6 mld snooping vlan 1`	Returns to privileged EXEC mode.

Configuring a Static Multicast Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure an IPv6 multicast address and member ports for a VLAN.

To add a Layer 2 port as a member of a multicast group, perform this procedure:

Procedure

	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 mld snooping vlan** *vlan-id* **static** *ipv6_multicast_address* **interface** *interface-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1 static 3333.0000.1111 interface gigabitethernet 0/1`	Configures a multicast group with a Layer 2 port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. The VLAN ID range is 1 to 1001 and 1006 to 4094. • *ipv6_multicast_address* is the 128-bit group IPv6 address. The address must be in the form specified in RFC 2373. • *interface-id* is the member port. It can be a physical interface or a port channel (1 to 48).
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 5	Use one of the following:
	• **show ipv6 mld snooping address**
	• **show ipv6 mld snooping address vlan** *vlan-id*
	Example:
	`Device# show ipv6 mld snooping address`
	or
	`Device# show ipv6 mld snooping vlan 1`
		Verifies the static member port and the IPv6 address.

Configuring a Multicast Router Port

Note Static connections to multicast routers are supported only on switch ports.

To add a multicast router port to a VLAN, perform this procedure:

Procedure

	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 mld snooping vlan** *vlan-id* **mrouter interface** *interface-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1 mrouter interface gigabitethernet 0/2`	Specifies the multicast router VLAN ID, and specify the interface to the multicast router. • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 48.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 mld snooping mrouter** [**vlan** *vlan-id*] **Example:** `Device# show ipv6 mld snooping mrouter vlan 1`	Verifies that IPv6 MLD snooping is enabled on the VLAN interface.

Enabling MLD Immediate Leave

To enable MLDv1 immediate leave, perform this procedure:

Procedure

	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	**ipv6 mld snooping vlan** *vlan-id* **immediate-leave** **Example:** `Device(config)# ipv6 mld snooping vlan 1 immediate-leave`	Enables MLD Immediate Leave on the VLAN interface.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 mld snooping vlan** *vlan-id* **Example:** `Device# show ipv6 mld snooping vlan 1`	Verifies that Immediate Leave is enabled on the VLAN interface.

Configuring MLD Snooping Queries

To configure MLD snooping query characteristics for the switch or for a VLAN, perform this procedure:

Procedure

	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 mld snooping robustness-variable** *value* **Example:** `Device(config)# ipv6 mld snooping robustness-variable 3`	(Optional) Sets the number of queries that are sent before switch will deletes a listener (port) that does not respond to a general query. The range is 1 to 3; the default is 2.
Step 4	**ipv6 mld snooping vlan** *vlan-id* **robustness-variable** *value* **Example:**	(Optional) Sets the robustness variable on a VLAN basis, which determines the number of general queries that MLD snooping sends before aging out a multicast address when there is no MLD report response. The range is 1 to 3; the

	Command or Action	Purpose
	`Device(config)# ipv6 mld snooping vlan 1 robustness-variable 3`	default is 0. When set to 0, the number used is the global robustness variable value.
Step 5	**ipv6 mld snooping last-listener-query-count** *count* **Example:** `Device(config)# ipv6 mld snooping last-listener-query-count 7`	(Optional) Sets the number of MASQs that the switch sends before aging out an MLD client. The range is 1 to 7; the default is 2. The queries are sent 1 second apart.
Step 6	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-count** *count* **Example:** `Device(config)# ipv6 mld snooping vlan 1 last-listener-query-count 7`	(Optional) Sets the last-listener query count on a VLAN basis. This value overrides the value configured globally. The range is 1 to 7; the default is 0. When set to 0, the global count value is used. Queries are sent 1 second apart.
Step 7	**ipv6 mld snooping last-listener-query-interval** *interval* **Example:** `Device(config)# ipv6 mld snooping last-listener-query-interval 2000`	(Optional) Sets the maximum response time that the switch waits after sending out a MASQ before deleting a port from the multicast group. The range is 100 to 32,768 thousands of a second. The default is 1000 (1 second).
Step 8	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-interval** *interval* **Example:** `Device(config)# ipv6 mld snooping vlan 1 last-listener-query-interval 2000`	(Optional) Sets the last-listener query interval on a VLAN basis. This value overrides the value configured globally. The range is 0 to 32,768 thousands of a second. The default is 0. When set to 0, the global last-listener query interval is used.
Step 9	**ipv6 mld snooping tcn query solicit** **Example:** `Device(config)# ipv6 mld snooping tcn query solicit`	(Optional) Enables topology change notification (TCN) solicitation, which means that VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it. The default is for TCN to be disabled.
Step 10	**ipv6 mld snooping tcn flood query count** *count* **Example:** `Device(config)# ipv6 mld snooping tcn flood query count 5`	(Optional) When TCN is enabled, specifies the number of TCN queries to be sent. The range is from 1 to 10; the default is 2.
Step 11	**end**	Returns to privileged EXEC mode.
Step 12	**show ipv6 mld snooping querier** [**vlan** *vlan-id*] **Example:** `Device(config)# show ipv6 mld snooping querier vlan 1`	(Optional) Verifies that the MLD snooping querier information for the switch or for the VLAN.

Disabling MLD Listener Message Suppression

MLD snooping listener message suppression is enabled by default. When it is enabled, the switch forwards only one MLD report per multicast router query. When message suppression is disabled, multiple MLD reports could be forwarded to the multicast routers.

To disable MLD listener message suppression, perform this procedure:

Procedure

	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`	Enter global configuration mode.
Step 3	**no ipv6 mld snooping listener-message-suppression** **Example:** `Device(config)# no ipv6 mld snooping listener-message-suppression`	Disable MLD message suppression.
Step 4	**end** **Example:** `Device(config)# end`	Return to privileged EXEC mode.
Step 5	**show ipv6 mld snooping** **Example:** `Device# show ipv6 mld snooping`	Verify that IPv6 MLD snooping report suppression is disabled.

Displaying MLD Snooping Information

You can display MLD snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display IPv6 group address multicast entries for a VLAN configured for MLD snooping.

Table 12: Commands for Displaying MLD Snooping Information

Command	Purpose
show ipv6 mld snooping [**vlan** *vlan-id*]	Displays the MLD snooping configuration information for all VLANs on the switch or for a specified VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping mrouter [**vlan** *vlan-id*]	Displays information on dynamically learned and manually configured multicast router interfaces. When you enable MLD snooping, the switch automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enters **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping querier [**vlan** *vlan-id*]	Displays information about the IPv6 address and incoming port for the most-recently received MLD query messages in the VLAN. (Optional) Enters **vlan** *vlan-id* to display information for a single VLAN.The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping address [**vlan** *vlan-id*] [**count** | **dynamic** | **user**]	Displays all IPv6 multicast address information or specific IPv6 multicast address information for the switch or a VLAN. • Enters **count** to show the group count on the switch or in a VLAN. • Enters **dynamic** to display MLD snooping learned group information for the switch or for a VLAN. • Enters **user** to display MLD snooping user-configured group information for the switch or for a VLAN.
show ipv6 mld snooping address vlan *vlan-id* [*ipv6-multicast-address*]	Displays MLD snooping for the specified VLAN and IPv6 multicast address.

Configuration Examples for Configuring MLD Snooping

Configuring a Static Multicast Group: Example

This example shows how to statically configure an IPv6 multicast group:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 2 static 3333.0000.1111 interface gigabitethernet1/0/1
Device(config)# end
```

Configuring a Multicast Router Port: Example

This example shows how to add a multicast router port to VLAN 200:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet

          0/2
Device(config)# exit
```

Enabling MLD Immediate Leave: Example

This example shows how to enable MLD Immediate Leave on VLAN 130:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 130 immediate-leave
Device(config)# exit
```

Configuring MLD Snooping Queries: Example

This example shows how to set the MLD snooping global robustness variable to 3:

```
Device# configure terminal
Device(config)# ipv6 mld snooping robustness-variable 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query count for a VLAN to 3:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query interval (maximum response time) to 2000 (2 seconds):

```
Device# configure terminal
Device(config)# ipv6 mld snooping last-listener-query-interval 2000
Device(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

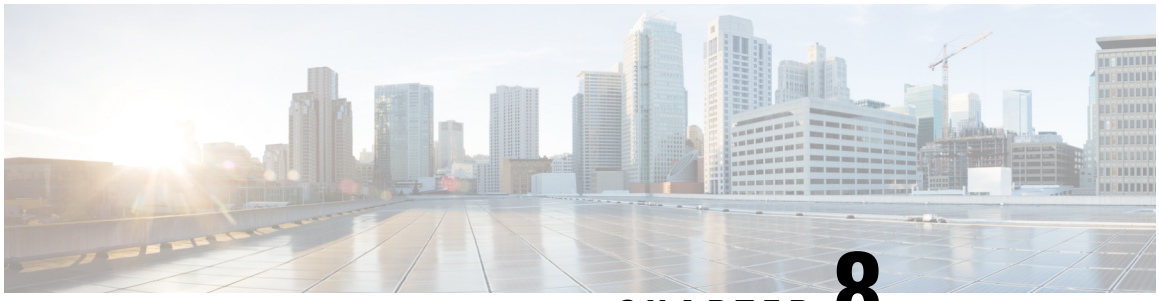
Feature Information for MLD Snooping

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 13: Feature Information for MLD Snooping

Feature Name	Releases	Feature Information
MLD Snooping	Cisco IOS XE Everest 16.5.1aCisco IOS XE Fuji 16.8.1a	MLD snooping allows the switch to examine MLD packets and make forwarding decisions based on their content. This feature was introduced for Cisco Catalyst 9500 Series Switches - High Performance



CHAPTER 8

Configuring IPv6 Support for LDAP

Restrictions for Configuring IPv6 Support for LDAP

- Only bind, search, and compare operations are supported.
- The Lightweight Directory Access Protocol (LDAP) referrals are not supported.
- Unsolicited messages or notifications from LDAP server are not handled.

Information About Configuring IPv6 Support for LDAP

IPv6 Support for LDAP

To support Lightweight Directory Access Protocol (LDAP) over IPv6, changes are made to authentication, authorization and accounting (AAA) transactions in terms of authentication and authorization while communicating over an IPv6 network. In order to support LDAP over an IPv6 network, transport calls have been modified to support both IPv4 and IPv6 based on the server configuration.

Transport Layer Security

Transport Layer Security (TLS) is an application-level protocol that enables secure transactions of data through privacy, authentication, and data integrity. It relies upon certificates, public keys, and private keys for clients to prove the identity. Certificates are issued by Certificate Authorities (CAs). Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date. TLS support for LDAP is mentioned in RFC 2830 as an extension to the LDAP protocol.

LDAP Operations

Bind

The bind operation is used to authenticate a user to the server. It is used to start a connection with the LDAP server. LDAP is a connection-oriented protocol. The client specifies the protocol version and the client authentication information. LDAP supports the following binds:

- Authenticated bind
- Anonymous bind

An authenticated bind is performed when a root distinguished name (DN) and password are available. In the absence of a root DN and password, an anonymous bind is performed. In LDAP deployments, the search operation is performed first and the bind operation later. This is because, if a password attribute is returned as part of the search operation, the password verification can be done locally on an LDAP client. Thus, there is no need to perform an extra bind operation. If a password attribute is not returned, the bind operation can be performed later. Another advantage of performing a search operation first and a bind operation later is that the DN received in the search result can be used as the user DN instead of forming a DN by prefixing the username (cn attribute) with the base DN. All entries stored in an LDAP server have a unique DN. The DN consists of two parts: the Relative Distinguished Name (RDN) and the location within the LDAP server where the record resides.

Most of the entries that you store in an LDAP server will have a name, and the name is frequently stored in the Common Name (cn) attribute. Because every object has a name, most objects you store in an LDAP will use their cn value as the basis for their RDN.

Compare

The compare operation is used to replace a bind request with a compare request for an authentication. The compare operation helps to maintain the initial bind parameters for the connection.

Search

A search operation is used to search the LDAP server. The client specifies the starting point (base DN) of the search, the search scope (either the object, its children, or the subtree rooted at the object), and a search filter.

For authorization requests, the search operation is directly performed without a bind operation. The LDAP server can be configured with certain privileges for the search operation to succeed. This privilege level is established with the bind operation.

An LDAP search operation can return multiple user entries for a specific user. In such cases, the LDAP client returns an appropriate error code to AAA. To avoid these errors, appropriate search filters that help to match a single entry must be configured.

How to Configure IPv6 Support for LDAP

Configuring Device-to-LDAP Server Communication

The Lightweight Directory Access Protocol (LDAP) host is a multiuser system running LDAP server software, such as Active Directory (Microsoft) and OpenLDAP. Configuring device-to-LDAP server communication can have several components:

- Hostname or IP address
- Port number
- Timeout period
- Base distinguished name (DN)

To configuring Device-to-LDAP server communication, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **ldap server** *name*
5. **ipv6** *ipv6-address*
6. **transport port** *port-number*
7. **timeout retransmit** *seconds*
8. **exit**

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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.

	Command or Action	Purpose
Step 4	**ldap server** *name* **Example:** `Device(config)# ldap server server1`	Configures a device as an LDAP protocol and enters LDAP server configuration mode.
Step 5	**ipv6** *ipv6-address* **Example:** `Device(config-ldap-server)# ipv6 2001:DB8:0:0:8:800`	Specifies an IPv6 address to the LDAP server.
Step 6	**transport port** *port-number* **Example:** `Device(config-ldap-server)# transport port 200`	Configures the transport protocol for connecting to the LDAP server.
Step 7	**timeout retransmit** *seconds* **Example:** `Device(config-ldap-server)# timeout retransmit 20`	Specifies the number of seconds a device waits for a reply to an LDAP request before retransmitting the request.
Step 8	**exit** **Example:** `Device(config-ldap-server)# exit`	Exits the LDAP server configuration mode and enters global configuration mode.

Configuring LDAP Protocol Parameters

To configure LDAP protocol parameters, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **ldap server** *name*
5. **bind authenticate root-dn password** [**0** *string* | **7** *string*] *string*
6. **search-filter user-object-type** *string*
7. **base-dn** *string*
8. **mode secure** [**no-negotiation**]
9. **secure cipher 3des-ede-cbc-sha**
10. **exit**

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	**aaa** **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	**ldap server** *name* **Example:** `Device(config)# ldap server server1`	Defines a Lightweight Directory Access Protocol (LDAP) server and enters LDAP server configuration mode.
Step 5	**bind authenticate root-dn password** [**0** *string* | **7** *string*] *string* **Example:** `Device(config-ldap-server)# bind authenticate root-dn "cn=administrator,cn=users,dc=nac-blr2,dc=example,dc=com password"`	Specifies a shared secret text string used between the device and an LDAP server. Use the **0** line option to configure an unencrypted shared secret. Use the **7** line option to configure an encrypted shared secret.
Step 6	**search-filter user-object-type** *string* **Example:** `Device(config-ldap-server)# search-filter user-object-type string1`	Specifies the search filter to be used in the search requests.
Step 7	**base-dn** *string* **Example:** `Device(config-ldap-server)# base-dn "dc=sns,dc=example,dc=com"`	Specifies the base distinguished name (DN) of the search.
Step 8	**mode secure** [**no-negotiation**] **Example:** `Device(config-ldap-server)# mode secure no-negotiation`	Configures LDAP to initiate the transport layer security (TLS) connection and specifies the secure mode.
Step 9	**secure cipher 3des-ede-cbc-sha** **Example:**	Specifies the ciphersuite in the case of a secure connection.

	Command or Action	Purpose
	`Device(config-ldap-server)# secure cipher 3des-ede-cbc-sha`	
Step 10	**exit** **Example:** `Device(config-ldap-server)# exit`	Exits LDAP server configuration mode and enters global configuration mode.

Configuring Search and Bind Operations for an Authentication Request

To configure search and bind operations for an authentication request, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **ldap server** *name*
5. **authentication bind-first**
6. **authentication compare**
7. **exit**

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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	**ldap server** *name* **Example:** `Device(config)# ldap server server1`	Defines a Lightweight Directory Access Protocol (LDAP) server and enter LDAP server configuration mode.

	Command or Action	Purpose
Step 5	**authentication bind-first** **Example:** `Device(config-ldap-server)# authentication bind-first`	Configures the sequence of search and bind operations for an authentication request.
Step 6	**authentication compare** **Example:** `Device(config-ldap-server)# authentication compare`	Replaces the bind request with the compare request for authentication.
Step 7	**exit** **Example:** `Device(config-ldap-server)# exit`	Exits LDAP server configuration mode.

Monitoring and Maintaining LDAP Scalability Enhancements

The following **show** and **debug** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clear ldap server**
4. **debug ldap**
5. **show ldap server**
6. **show ldap attributes**

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 **clear ldap server**

Clears the Lightweight Directory Access Protocol (LDAP) server of the TCP connection.

Example:

```
Device# clear ldap server
```

Step 4 **debug ldap**

Displays information associated with LDAP.

Example:

```
Device# debug ldap
```

Step 5 **show ldap server**

Displays the LDAP server state information and various other counters for the server.

Example:

```
Device# show ldap server
```

Step 6 **show ldap attributes**

Displays information about default LDAP attribute mapping.

Example:

```
Device# show ldap attributes

LDAP Attribute                  Format      AAA Attribute
==============                  ======      =============
airespaceBwDataBurstContract    Ulong       bsn-data-bandwidth-burst-contr
userPassword                    String      password
airespaceBwRealBurstContract    Ulong       bsn-realtime-bandwidth-burst-c
employeeType                    String      employee-type
airespaceServiceType            Ulong       service-type
airespaceACLName                String      bsn-acl-name
priv-lvl                        Ulong       priv-lvl
memberOf                        String DN   supplicant-group
cn                              String      username
airespaceDSCP                   Ulong       bsn-dscp
policyTag                       String      tag-name
airespaceQOSLevel               Ulong       bsn-qos-level
airespace8021PType              Ulong       bsn-8021p-type
airespaceBwRealAveContract      Ulong       bsn-realtime-bandwidth-average
airespaceVlanInterfaceName      String      bsn-vlan-interface-name
airespaceVapId                  Ulong       bsn-wlan-id
airespaceBwDataAveContract      Ulong       bsn-data-bandwidth-average-con
sAMAccountName                  String      sam-account-name
meetingContactInfo              String      contact-info
telephoneNumber                 String      telephone-number
Map: att_map_1
department                      String DN   element-req-qos
```

Configuration Examples of IPv6 Support for LDAP

Example: Device-to-LDAP Server Communication

The following example shows how to create server group server1 and specify the IP address, transport port 200, and retransmit values:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# ldap server server1
Device(config-ldap-server)# ipv6 2001:DB8:0:0:8:800
Device(config-ldap-server)# transport port 200
Device(config-ldap-server)# timeout retransmit 20
Device(config-ldap-server)# exit
```

Example: LDAP Protocol Parameters

The following example shows how to configure Lightweight Directory Access Protocol (LDAP) parameters:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# ldap server server1
Device(config-ldap-server)# bind authenticate root-dn
"cn=administrator,cn=users,dc=nac-blr2,dc=example,dc=com password"
Device(config-ldap-server)# base-dn "dc=sns,dc=example,dc=com"
Device(config-ldap-server)# mode secure no-negotiation
Device(config-ldap-server)# secure cipher 3des-ede-cbc-sha
Device(config-ldap-server)# exit
```

Example: Search and Bind Operations for an Authentication Request

The following example shows how to configure the sequence of search and bind operations for an authentication request:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# ldap server server1
Device(config-ldap-server)# authentication bind-first
Device(config-ldap-server)# authentication compare
Device(config-ldap-server)# exit
```

Example: Server Information from an LDAP Server

The following is sample output from an LDAP server:

```
Device# show ldap server all

Server Information for server1
```

```
==========================================
 Server name              :server1
 Server IP                :2001:DB8:0:0:8:800
 Server listening Port    :389
 Connection status        :DOWN
 Root Bind status         :No Bind
 Server mode              :Non-Secure
 Cipher Suite             :0x00
 Authentication Seq       :Search first. Then Bind/Compare      password next
 Authentication Procedure :Bind with user password
 Request timeout          :30
 ------------------------------------------
 * LDAP STATISTICS *
 Total messages  [Sent:0, Received:0]
 Response delay(ms) [Average:0, Maximum:0]
 Total search    [Request:0, ResultEntry:0, ResultDone:0]
 Total bind      [Request:0, Response:0]
 Total extended  [Request:0, Response:0]
 Total compare   [Request:0, Response:0]
 Search [Success:0, Failures:0]
 Bind   [Success:0, Failures:0]
 Missing attrs in Entry [0]
------------------------------------------
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

Standards and RFCs

Standard/RFC	Title
RFC 4511	*Lightweight Directory Access Protocol (LDAP)*
RFC 4513	*Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms*

Feature History for IPv6 Support for LDAP

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.8.1a	IPv6 Support for LDAP	The IPv6 Support for LDAP feature describes IPv6 transport support for the LDAP protocol by introducing changes in authentication, authorization, and accounting (AAA) transactions. Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER 9

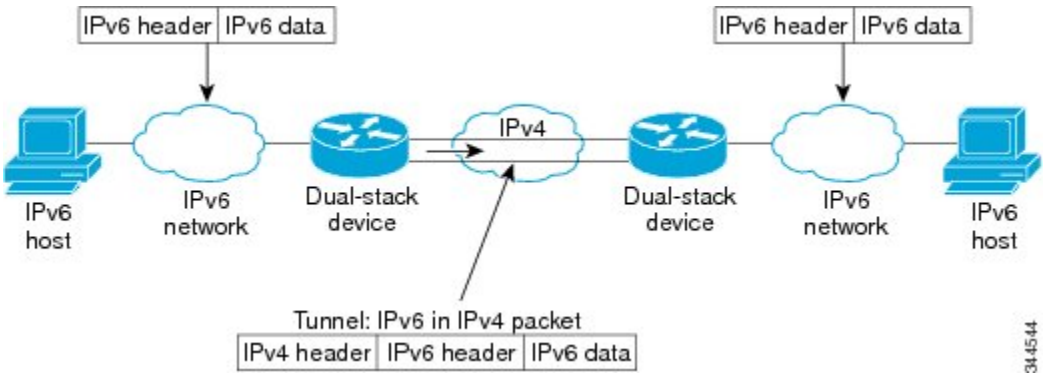
Configuring IPv6 over IPv4 GRE Tunnels

Information About Configuring IPv6 over IPv4 GRE Tunnels

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the figure below). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border devices or between a border device and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks.

Figure 5: Overlay Tunnels



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

IPv6 supports GRE type of overlay tunneling. IPv6 over IPv4 GRE Tunnels can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.

GRE IPv4 Tunnel Support for IPv6 Traffic

IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol but, in this case, carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.

The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge devices or between an edge device and an end system. The edge devices and the end systems must be dual-stack implementations.

How to Configure IPv6 over IPv4 GRE Tunnels

Configuring GRE IPv6 Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and to transport IPv6 packets in IPv6 tunnels and IPv4 packets in IPv6 tunnels.

To configure GRE IPv6 tunnels, perform this procedure:

Before you begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses assigned (this is not shown in the task). The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
6. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}

7. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre**| **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos**

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 tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Device(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source ethernet 0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. • If an interface is specified, the interface must be configured with an IPv4 address.
Step 6	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Device(config-if)# tunnel destination 2001:DB8:1111:2222::1/64`	Specifies the destination IPv6 address or hostname for the tunnel interface.
Step 7	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre**| **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6**command specifies GRE as the encapsulation protocol for the tunnel.

Configuration Examples for IPv6 over IPv4 GRE Tunnels

Example: GRE Tunnel Running IS-IS and IPv6 Traffic

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between Router A and Router B:

Router A Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::3/127
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 2001:DB8:1111:2222::1/64
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 net 49.0000.0000.000a.00
```

Router B Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::2/127
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 2001:DB8:1111:2222::2/64
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 net 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Example: Tunnel Destination Address for IPv6 Tunnel

```
interface Tunnel 0
ipv6 address 2001:1:1::1/48
tunnel source GigabitEthernet 0/0/0
tunnel destination 10.0.0.2
```

```
tunnel mode gre ipv6
exit
!
interface GigabitEthernet0/0/0
ip address 10.0.0.1 255.255.255.0
exit
!
ipv6 unicast-routing
router isis
net 49.0000.0000.000a.00



Router(config
)
#
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

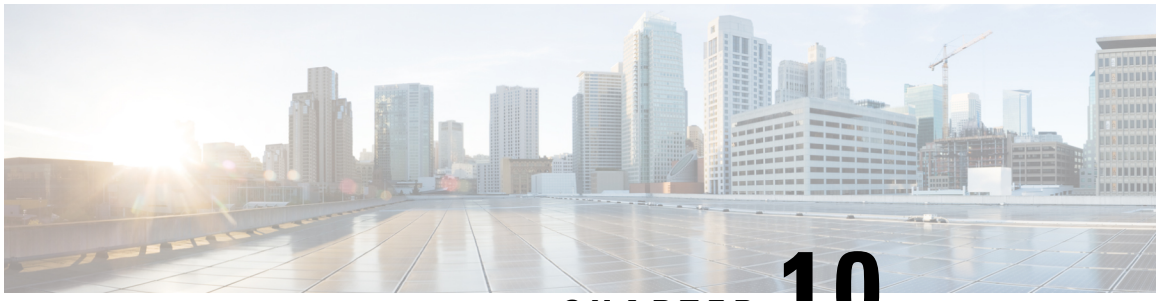
Feature Information

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 14: Feature Information for IPv6 over IPv4 GRE Tunnels

Feature Name	Releases	Feature Information
IPv6 over IPv4 GRE Tunnels	Cisco IOS XE Fuji 16.8.1a	GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.



CHAPTER 10

Configuring IPv6 ACL

Prerequisites for Configuring IPv6 ACL

You can filter IP Version 6 (IPv6) traffic by creating IPv6 access control lists (ACLs) and applying them to interfaces similarly to the way that you create and apply IP Version 4 (IPv4) named ACLs. You can also create and apply input router ACLs to filter Layer 3 management traffic when the switch is running the Network Essentials license.

Restrictions for Configuring IPv6 ACL

With IPv4, you can configure standard and extended numbered IP ACLs, named IP ACLs, and MAC ACLs. IPv6 supports only named ACLs.

The device supports most of the Cisco IOS-supported IPv6 ACLs with some exceptions:

- The device does not support matching on these keywords: **flowlabel**, **routing header**, and **undetermined-transport**.
- The device does not support reflexive ACLs (the **reflect** keyword).
- The device does not apply MAC-based ACLs on IPv6 frames.
- When configuring an ACL, there is no restriction on keywords entered in the ACL, regardless of whether or not they are supported on the platform. When you apply the ACL to an interface that requires hardware

forwarding (physical ports or SVIs), the device checks to determine whether or not the ACL can be supported on the interface. If not, attaching the ACL is rejected.

- If an ACL is applied to an interface and you attempt to add an access control entry (ACE) with an unsupported keyword, the device does not allow the ACE to be added to the ACL that is currently attached to the interface

Information About IPv6 ACL

An access control list (ACL) is a set of rules used to limit access to a particular interface. ACLs are configured on the device and applied to the management interface and to any of the dynamic interfaces.

You can also create a preauthentication ACL for web authentication. Such an ACL is used to allow certain types of traffic before authentication is complete.

IPv6 ACLs support the same options as IPv4 ACLs including source, destination, source and destination ports.

Note You can enable only IPv4 traffic in your network by blocking IPv6 traffic. That is, you can configure an IPv6 ACL to deny all IPv6 traffic and apply it on specific or all WLANs.

Understanding IPv6 ACLs

A switch supports two types of IPv6 ACLs:

- IPv6 router ACLs are supported on outbound or inbound traffic on Layer 3 interfaces, which can be routed ports, switch virtual interfaces (SVIs), or Layer 3 EtherChannels. IPv6 router ACLs apply only to IPv6 packets that are routed.
- IPv6 port ACLs are supported on inbound traffic on Layer 2 interfaces only. IPv6 port ACLs are applied to all IPv6 packets entering the interface.

A switch running the Network Essentials license supports only input router IPv6 ACLs. It does not support port ACLs or output IPv6 router ACLs.

Note If you configure unsupported IPv6 ACLs, an error message appears and the configuration does not take affect.

The switch does not support VLAN ACLs (VLAN maps) for IPv6 traffic.

You can apply both IPv4 and IPv6 ACLs to an interface. As with IPv4 ACLs, IPv6 port ACLs take precedence over router ACLs:

- When an input router ACL and input port ACL exist in an SVI, packets received on ports to which a port ACL is applied are filtered by the port ACL. Routed IP packets received on other ports are filtered by the router ACL. Other packets are not filtered.
- When an output router ACL and input port ACL exist in an SVI, packets received on the ports to

  which a port ACL is applied are filtered by the port ACL. Outgoing routed IPv6 packets are filtered by the router ACL. Other packets are not filtered.

Note If any port ACL (IPv4, IPv6, or MAC) is applied to an interface, that port ACL is used to filter packets, and any router ACLs attached to the SVI of the port VLAN are ignored.

Types of ACL

Per User IPv6 ACL

For the per-user ACL, the full access control entries (ACE) as the text strings are configured on the Cisco Secure Access Control Server (Cisco Secure ACS).

Filter ID IPv6 ACL

For the filter-Id ACL, the full ACEs and the `acl name(filter-id)` is configured on the device and only the `filter-id` is configured on the Cisco Secure ACS.

IPv6 ACLs and Switch Stacks

The active switch supports IPv6 ACLs in hardware and distributes the IPv6 ACLs to the stack's member switches.

Note For full IPv6 functionality in a switch stack, all member switches must be running the Network Advantage license.

If a new switch takes over as active switch, it distributes the ACL configuration to all member switches. The member switches sync up the configuration distributed by the new active switch and flush out entries that member switches sync up the configuration distributed by the new active switch and flush out entries that are not required.

When an ACL is modified, attached to, or detached from an interface, the active switch distributes the change to all member switches.

Configuring IPv6 ACLs

Follow the procedure given below to filter IPv6 traffic:

1. Create an IPv6 ACL, and enter IPv6 access list configuration mode.
2. Configure the IPv6 ACL to block (deny) or pass (permit) traffic.
3. Apply the IPv6 ACL to the interface where the traffic needs to be filtered.
4. Apply the IPv6 ACL to an interface. For router ACLs, you must also configure an IPv6 address on the Layer 3 interface to which the ACL is applied.

Before you begin

Before configuring IPv6 ACLs, you must select one of the IPv6 SDM templates.

Default IPv6 ACL Configuration

There are no IPv6 ACLs configured or applied.

Interaction with Other Features and Switches

- If an IPv6 router ACL is configured to deny a packet, the packet is not routed. A copy of the packet is sent to the Internet Control Message Protocol (ICMP) queue to generate an ICMP unreachable message for the frame.
- If a bridged frame is to be dropped due to a port ACL, the frame is not bridged.
- You can create both IPv4 and IPv6 ACLs on a switch or switch stack, and you can apply both IPv4 and IPv6 ACLs to the same interface. Each ACL must have a unique name; an error message appears if you try to use a name that is already configured.

  You use different commands to create IPv4 and IPv6 ACLs and to attach IPv4 or IPv6 ACLs to the same Layer 2 or Layer 3 interface. If you use the wrong command to attach an ACL (for example, an IPv4 command to attach an IPv6 ACL), you receive an error message.

- You cannot use MAC ACLs to filter IPv6 frames. MAC ACLs can only filter non-IP frames.
- If the hardware memory is full, for any additional configured ACLs, packets are dropped to the CPU, and the ACLs are applied in software. When the hardware is full a message is printed to the console indicating the ACL has been unloaded and the packets will be dropped on the interface.

How To Configure an IPv6 ACL

Creating an IPv6 ACL

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *acl_name*
4. **{deny|permit} protocol**
5. **{deny|permit} tcp**
6. **{deny|permit} udp**
7. **{deny|permit} icmp**
8. **end**
9. **show ipv6 access-list**
10. **copy running-config startup-config**

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 access-list** *acl_name* **Example:** `Device# ipv6 access-list access-list-name`	Use a name to define an IPv6 access list and enter IPv6 access-list configuration mode.
Step 4	**{deny|permit} protocol** **Example:** `{deny | permit} protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length | any |host destination-ipv6-address} [operator [port-number]][dscp value] [fragments][log] [log-input] [routing][sequence value] [time-range name]`	Enter deny or permit to specify whether to deny or permit the packet if conditions are matched. These are the conditions: • For protocol, enter the name or number of an Internet protocol: ahp, esp, icmp, ipv6, pcp, stcp, tcp, or udp, or an integer in the range 0 to 255 representing an IPv6 protocol number. • The source-ipv6-prefix/prefix-length or destination-ipv6-prefix/ prefix-length is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373). • Enter any as an abbreviation for the IPv6 prefix ::/0. • For host source-ipv6-address or destination-ipv6-address, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons. • (Optional) For operator, specify an operand that compares the source or destination ports of the specified protocol. Operands are lt (less than), gt (greater than), eq (equal), neq (not equal), and range. If the operator follows the source-ipv6-prefix/prefix-length argument, it must match the source port. If the operator follows the destination-ipv6- prefix/prefix-length argument, it must match the destination port.

	Command or Action	Purpose
		• (Optional) The port-number is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP. • (Optional) Enter dscp value to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63. • (Optional) Enter fragments to check noninitial fragments. This keyword is visible only if the protocol is ipv6. • (Optional) Enter log to cause an logging message to be sent to the console about the packet that matches the entry. Enter log-input to include the input interface in the log entry. Logging is supported only for router ACLs. • (Optional) Enter routing to specify that IPv6 packets be routed. • (Optional) Enter sequence value to specify the sequence number for the access list statement. The acceptable range is from 1 to 4294967295 • (Optional) Enter time-range name to specify the time range that applies to the deny or permit statement.
Step 5	**{deny|permit} tcp** **Example:** `{deny | permit} tcp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length | any |hostdestination-ipv6-address} [operator [port-number]][ack] [dscp value][established] [fin] [log][log-input] [neq {port |protocol}] [psh] [range{port | protocol}] [rst][routing] [sequence value] [syn] [time-range name][urg]`	(Optional) Define a TCP access list and the access conditions. Enter tcp for Transmission Control Protocol. The parameters are the same as those described in Step 3, with these additional optional parameters: • ack—Acknowledgment bit set. • established—An established connection. A match occurs if the TCP datagram has the ACK or RST bits set. • fin—Finished bit set; no more data from sender. • neq {port | protocol}—Matches only packets that are not on a given port number. • psh—Push function bit set. • range {port | protocol}—Matches only packets in the port number range. • rst—Reset bit set.

	Command or Action	Purpose
		• syn—Synchronize bit set. • urg—Urgent pointer bit set.
Step 6	**{deny|permit} udp** **Example:** `{deny | permit} udp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length | any | hostdestination-ipv6-address} [operator [port-number]][dscp value] [log][log-input] [neq {port |protocol}] [range {port |protocol}] [routing][sequence value][time-range name]`	(Optional) Define a UDP access list and the access conditions. Enter udp for the User Datagram Protocol. The UDP parameters are the same as those described for TCP, except that the operator [port]] port number or name must be a UDP port number or name, and the established parameter is not valid for UDP.
Step 7	**{deny|permit} icmp** **Example:** `{deny | permit} icmp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | hostdestination-ipv6-address} [operator [port-number]][icmp-type [icmp-code] |icmp-message] [dscpvalue] [log] [log-input] [routing] [sequence value][time-range name]`	(Optional) Define an ICMP access list and the access conditions. Enter icmp for Internet Control Message Protocol. The ICMP parameters are the same as those described for most IP protocols in Step 3a, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: • icmp-type—Enter to filter by ICMP message type, a number from 0 to 255. • icmp-code—Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255. • icmp-message—Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name. To see a list of ICMP message type names and code names, use the ? key or see command reference for this release.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.
Step 9	**show ipv6 access-list** **Example:** `show ipv6 access-list`	Verify the access list configuration.
Step 10	**copy running-config startup-config** **Example:** `copy running-config startup-config`	(Optional) Save your entries in the configuration file.

Applying an IPv6 to an Interface

This section describes how to apply IPv6 ACLs to network interfaces. You can apply an IPv6 ACL to outbound or inbound traffic on layer 2 and Layer 3 interfaces. You can apply IPv6 ACLs only to inbound management traffic on Layer 3 interfaces.

To control access to an interface, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface_id*
4. **no switchport**
5. **ipv6 address** *ipv6_address*
6. **ipv6 traffic-filter** *acl_name*
7. **end**
8. **show running-config interface tenGigabitEthernet 1/0/3**
9. **copy running-config startup-config**

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** *interface_id* **Example:** `Device# interface interface-id`	Identifies a Layer 2 interface (for port ACLs) or Layer 3 Switch Virtual interface (for router ACLs) on which to apply an access list, and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device# no switchport`	Changes the interface from Layer 2 mode (the default) to Layer 3 mode (only if applying a router ACL).
Step 5	**ipv6 address** *ipv6_address* **Example:** `Device# ipv6 address ipv6-address`	Configures an IPv6 address on a Layer 3 interface (for router ACLs). **Note** This command is not required on Layer 2 interfaces or if the interface has already been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 6	**ipv6 traffic-filter** *acl_name* **Example:** `Device# ipv6 traffic-filter access-list-name {in | out}`	Applies the access list to incoming or outgoing traffic on the interface.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Step 8	**show running-config interface tenGigabitEthernet 1/0/3**
	Example:
	`Device# show running-config interface tenGigabitEthernet 1/0/3`
	`.....`
	`.....`
	`Building configuration .....`
	`.....`
	`Current configuration : 98 bytes`
	`!`
	`interface TenGigabitEthernet1/0/3`
	` switchport mode trunk`
	` ipv6 traffic-filter MyFilter out`
	`end`
		Shows the configuration summary.

Step 9	**copy running-config startup-config** **Example:** `copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Verifying IPv6 ACL

Displaying IPv6 ACLs

To display IPv6 ACLs, perform this procedure:

Procedure

	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	**show access-list** **Example:** `Device# show access-lists`	Displays all access lists configured on the device
Step 4	**show ipv6 access-list** *acl_name* **Example:** `Device# show ipv6 access-list [access-list-name]`	Displays all configured IPv6 access list or the access list specified by name.

Configuring RA Guard Policy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd raguard policy** *policy name*
4. **trusted-port**
5. **device-role router**
6. **interface** *interface-id*
7. **ipv6 nd raguard attach-policy** *policy name*
8. **vlan** *vlan-id*
9. **ipv6 nd suppress**
10. **ipv6 snooping**
11. **ipv6 nd raguard attach-policy** *policy name*
12. **ipv6 nd ra-throttler attach-policy** *policy 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	**ipv6 nd raguard policy** *policy name* **Example:** `Device(config)# ipv6 nd raguard policy MyPolicy`	

	Command or Action	Purpose
Step 4	**trusted-port** **Example:** `Device(config-nd-raguard)# trusted-port`	Configures the trusted port for the policy created above.

Step 5	**device-role router**
	Example:
	`Device(config-nd-raguard)# device-role [host|monitor|router|switch]`
	`Device(config-nd-raguard)# device-role router d`
		Defines the trusted device that can send RAs to the trusted port created above.

Step 6	**interface** *interface-id* **Example:** `Device(config)# interface tenGigabitEthernet 1/0/1`	Configures the interface to the trusted device.
Step 7	**ipv6 nd raguard attach-policy** *policy name* **Example:** `Device(config-if)# ipv6 nd raguard attach-policy Mypolicy`	Configures and attaches the policy to trust the RA's received from the port.
Step 8	**vlan** *vlan-id* **Example:** `Device(config)# vlan configuration 19-21,23`	Configures the wireless client vlans.
Step 9	**ipv6 nd suppress** **Example:** `Device(config-vlan-config)# ipv6 nd suppress`	Suppresses the ND messages over wireless.
Step 10	**ipv6 snooping** **Example:** `Device(config-vlan-config)# ipv6 snooping`	Captures IPv6 traffic.
Step 11	**ipv6 nd raguard attach-policy** *policy name* **Example:** `Device(config-vlan-config)# ipv6 nd raguard attach-policy Mypolicy`	Attaches the RA Guard policy to the wireless client vlans.
Step 12	**ipv6 nd ra-throttler attach-policy** *policy name* **Example:** `Device(config-vlan-config)# ipv6 nd ra-throttler attach-policy Mythrottle`	Attaches the RA throttling policy to the wireless client vlans.

Configuring IPv6 Neighbor Binding

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor binding** [vlan] `19 2001:db8::25:4` **interface tenGigabitEthernet** `1/0/3 aaa.bbb.ccc`

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 neighbor binding** [vlan] `19 2001:db8::25:4` **interface tenGigabitEthernet** `1/0/3 aaa.bbb.ccc` **Example:** `Device(config)# ipv6 neighbor binding vlan 19 2001:db8::25:4 interface tenGigabitEthernet 1/0/3 aaa.bbb.ccc`	Sets and validates the neighbor 2001:db8::25: 4 only valid when transmitting on VLAN 19 through interface te1/0/3 with the source mac-address as aaa.bbb.ccc.

Configuration Examples for IPv6 ACL

Example: Creating an IPv6 ACL

This example configures the IPv6 access list named CISCO. The first deny entry in the list denies all packets that have a destination TCP port number greater than 5000. The second deny entry denies packets that have a source UDP port number less than 5000. The second deny also logs all matches to the console. The first permit entry in the list permits all ICMP packets. The second permit entry in the list permits all other traffic. The second permit entry is necessary because an implicit deny -all condition is at the end of each IPv6 access list.

Note Logging is supported only on Layer 3 interfaces.

```
Device(config)# ipv6 access-list CISCO
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device (config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
```

Example: Applying IPv6 ACLs

This example shows how to apply the access list Cisco to outbound traffic on a Layer 3 interface.

```
Device(config)# interface TenGigabitEthernet 1/0/3

Device(config-if)# no switchport
Device(config-if)# ipv6 address 2001::/64 eui-64
Device(config-if)# ipv6 traffic-filter CISCO out
```

Example: Displaying IPv6 ACLs

This is an example of the output from the **show access-lists** privileged EXEC command. The output shows all access lists that are configured on the switch or switch stack.

```
Device #show access-lists
Extended IP access list hello
10 permit ip any any
IPv6 access list ipv6
permit ipv6 any any sequence 10
```

This is an example of the output from the show ipv6 access-lists privileged EXEC command. The output shows only IPv6 access lists configured on the switch or switch stack.

```
Device# show ipv6 access-list
IPv6 access list inbound
permit tcp any any eq bgp (8 matches) sequence 10
permit tcp any any eq telnet (15 matches) sequence 20
permit udp any any sequence 30

IPv6 access list outbound
deny udp any any sequence 10
deny tcp any any eq telnet sequence 20
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9500 Series Switches)*

Feature Information for IPv6 ACLs

This table lists the features in this module and provides links to specific configuration information:

Feature	Release	Modification
IPv6 ACL Functionality	Cisco IOS XE Everest 16.5.1aCisco IOS XE Fuji 16.8.1a	This feature was introduced. This feature was introduced for Cisco Catalyst 9500 Series Switches - High Performance