



CHAPTER 32

Configuring IPv6 Host Functions

**Note**

To use IPv6 Host Functions, the switch must be running the LAN Base image.

Internet Protocol Version 6 (IPv6) is the network-layer Internet Protocol intended to replace Version 4 (IPv4) in the TCP/IP suite of protocols. This chapter describes how to configure IPv6 host functions on the Catalyst 2960 switch.

For information about configuring IPv6 Multicast Listener Discovery (MLD) snooping, see [Chapter 33, “Configuring IPv6 MLD Snooping.”](#)

To enable dual stack environments (supporting both IPv4 and IPv6), you must configure a switch database management (SDM) template to a dual IPv4 and IPv6 template. See the [“SDM Templates” section on page 32-12](#).

**Note**

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

This chapter consists of these sections:

- [“Understanding IPv6” section on page 32-1](#)
- [“Configuring IPv6” section on page 32-13](#)
- [“Displaying IPv6” section on page 32-17](#)

Understanding IPv6

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

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

IPv6 also provides these advantages over IPv4:

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

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

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

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

- [IPv6 Addresses, page 32-2](#)
- [Supported IPv6 Unicast Routing Features, page 32-3](#)
- [SDM Templates, page 32-12](#)

IPv6 Addresses

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

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

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

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

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

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

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

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

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

In the “Implementing Addressing and Basic Connectivity” chapter, these sections apply to the Catalyst 2960 switch:

- IPv6 Address Formats
- IPv6 Address Output Display
- Simplified IPv6 Packet Header

Supported IPv6 Unicast Routing Features

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

- [128-Bit Wide Unicast Addresses, page 32-3](#)
- [ICMPv6, page 32-4](#)
- [Neighbor Discovery, page 32-4](#)
- [IPv6 Stateless Autoconfiguration and Duplicate Address Detection, page 32-4](#)
- [IPv6 Applications, page 32-9](#)
- [Dual IPv4 and IPv6 Protocol Stacks, page 32-10](#)
- [SNMP and Syslog Over IPv6, page 32-10](#)
- [HTTP\(S\) Over IPv6, page 32-11](#)

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

128-Bit Wide Unicast Addresses

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

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

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

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

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

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

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

DNS for IPv6

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

ICMPv6

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

Neighbor Discovery

The switch supports Neighbor Discovery Protocol (NDP) for IPv6 (RFC 2461), a protocol running on top of ICMPv6, and Static Neighbor Discovery for IPv6 stations that do not support NDP. The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of the neighbor, and keep track of neighboring routers.

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

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

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

IPv6 Stateless Autoconfiguration and Duplicate Address Detection

IPv6 supports two types of autoconfiguration:

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

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

Every interface on IPv6 nodes must have a link-local address that is automatically configured from the identifier (router MAC address) for an interface and the link-local prefix (FE80::/10). A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node. Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or the help of a server such as a DHCP server.

With IPv6, a router on the link uses router advertisement messages to advertise global prefixes and its ability to act as a default router for the link. A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the router advertisement messages.

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

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

IPv6 Stateless Autoconfiguration

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

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

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

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

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

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

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

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

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

Link-local Address Configuration

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

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

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

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

switch# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
No global unicast address is configured
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF2E:9047
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Global-Address Configuration

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

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

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

This is the configuration on the router side:

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

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

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

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

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

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

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

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

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

```

ipv6 address 1016:2::1/72
end

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

```

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

```

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

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

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

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

switch2# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
Global unicast address(es):
  1016:1::20B:46FF:FE2E:9047, subnet is 1016:1::/64 [PRE]
    valid lifetime 2591999 preferred lifetime 604799
  1016:2::20B:46FF:FE2E:9047, subnet is 1016:2::/64 [PRE/TEN]
    valid lifetime 179 preferred lifetime 179
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF2E:9047
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.

```

This is a site-local address configuration:

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

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

This is the configuration on the router side:

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

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

switch# show ipv6 interface fastethernet1/0/16
FastEthernet1/0/16 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::20B:46FF:FE2E:9047
 Global unicast address(es):
   FEC0:1016:1:0:20B:46FF:FE2E:9047, subnet is FEC0:1016:1::/64 [PRE]
   valid lifetime 2591834 preferred lifetime 604634
 Joined group address(es):
   FF02::1
   FF02::2
   FF02::1:FF2E:9047
 MTU is 1500 bytes
 ICMP error messages limited to one every 100 milliseconds
 ICMP redirects are enabled
 ND DAD is enabled, number of DAD attempts: 1
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND router advertisements are sent every 200 seconds
 ND router advertisements live for 1800 seconds
 Hosts use stateless autoconfig for addresses.
```

IPv6 Applications

The switch has IPv6 support for these applications:

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

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

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

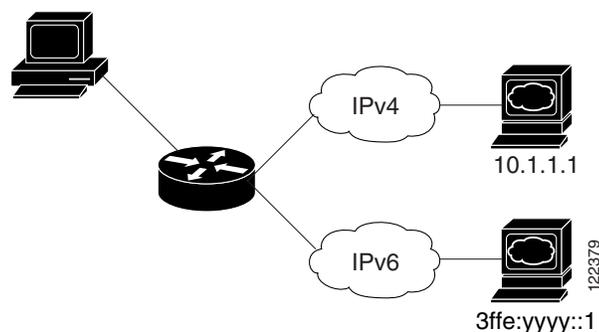
Dual IPv4 and IPv6 Protocol Stacks

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

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

Figure 32-1 shows a router forwarding both IPv4 and IPv6 traffic through the same interface, based on the IP packet and destination addresses.

Figure 32-1 Dual IPv4 and IPv6 Support on an Interface



The switch uses ternary content addressable memory (TCAM) to store unicast routes, MAC addresses, access control lists (ACLs), and other features, and provides the switch database management (SDM) templates to allocate memory resources depending on how the switch is used. You must use the dual IPv4 and IPv6 template templates to allocate TCAM usage to both IPv4 and IPv6 protocols. See the “[SDM Templates](#)” section on page 32-12.

SNMP and Syslog Over IPv6

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

SNMP and syslog over IPv6 provides these features:

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

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

- Opens User Datagram Protocol (UDP) SNMP socket with default settings
- Provides a new transport mechanism called, *SR_IPV6_TRANSPORT*
- Sends SNMP notifications over IPv6 transport
- Supports SNMP-named access lists for IPv6 transport
- Supports SNMP proxy forwarding using IPv6 transport
- Verifies SNMP Manger feature works with IPv6 transport

For information on SNMP over IPv6, see:

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

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

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

Syslog Over IPv6

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

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

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

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

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

Commands

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

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

HTTP(S) Over IPv6

The HTTP client in Cisco IOS supports sending requests to both IPv4 and IPv6 HTTP servers. The HTTP server in Cisco IOS can service HTTP requests from both IPv4 and IPv6 HTTP clients. URLs with literal

IPv6 addresses must be formatted by using the rules listed in RFC 2732.

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

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

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

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

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

SDM Templates

Catalyst 2960 switches have one TCAM to store unicast routes, MAC addresses, ACLs, and other features. To allocate TCAM resources for different usages, the switch SDM templates prioritize system resources to optimize support for certain features. You select the template that best suits the switch environment by entering the **sdm prefer** global configuration command. For more information about SDM templates, see [Chapter 7, “Configuring SDM Templates.”](#)

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



Note

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

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



Note

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

Dual IPv4-and-IPv6 SDM Templates

These SDM templates support IPv4 and IPv6 environments:



Note

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

Dual IPv4 and IPv6 default SDM template—supports Layer 2, QoS, and ACLs for IPv4; and Layer 2 and ACLs for IPv6 on the switch.

**Note**

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

Table 32-1 defines the approximate feature resources allocated by each new template.

Table 32-1 *Approximate Number of Feature Resources Allowed by Each Template*

Resource	Default	QoS	Dual
Unicast MAC addresses	8 K	8 K	8 K
IPv4 IGMP groups + multicast routes	.25 K	.25 K	.25 K
IPv4 unicast routes	0	0	0
IPv6 multicast groups	0	0	.25 K
Directly connected IPv6 addresses	0	0	0
Indirect IPv6 unicast routes	0	0	0
IPv4 policy-based routing aces	0	0	0
IPv4 MAC QoS ACEs	128	384	.25 K
IPv4 MAC security ACEs	384	128	.25 K
IPv6 policy-based routing aces	0	0	0
IPv4 MAC QoS ACEs	0	0	0
IPv4 MAC security ACEs	0	0	0

Configuring IPv6

These sections contain this IPv6 forwarding configuration information:

- [Default IPv6 Configuration, page 32-13](#)
- [Configuring IPv6 ICMP Rate Limiting, page 32-14](#)
- [Configuring Static Routes for IPv6, page 32-14](#)

Default IPv6 Configuration

Table 32-2 shows the default IPv6 configuration.

Table 32-2 *Default IPv6 Configuration*

Feature	Default Setting
SDM template	Default
IPv6 addresses	None configured

Configuring IPv6 ICMP Rate Limiting

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

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

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

	Command	Purpose
Step 1	<code>configure terminal</code>	Enter global configuration mode.
Step 2	<code>ipv6 icmp error-interval interval [bucketsize]</code>	Configure the interval and bucket size for IPv6 ICMP error messages: <ul style="list-style-type: none"> <i>interval</i>—The interval (in milliseconds) between tokens being added to the bucket. The range is from 0 to 2147483647 milliseconds. <i>bucketsize</i>—(Optional) The maximum number of tokens stored in the bucket. The range is from 1 to 200.
Step 3	<code>end</code>	Return to privileged EXEC mode.
Step 4	<code>show ipv6 interface [interface-id]</code>	Verify your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Save your entries in the configuration file.

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

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

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

Configuring Static Routes for IPv6

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

There are types of static routes:

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

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

Command	Purpose
Step 1 configure terminal	Enter global configuration mode.
Step 2 ipv6 route <i>ipv6-prefix/prefix length</i> { <i>ipv6-address</i> <i>interface-id</i> [<i>ipv6-address</i>]} [<i>administrative distance</i>]	Configure a static IPv6 route. <ul style="list-style-type: none"> • <i>ipv6-prefix</i>—The IPv6 network that is the destination of the static route. It can also be a hostname when static host routes are configured. • <i>/prefix length</i>—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. • <i>ipv6-address</i>—The IPv6 address of the next hop that can be used to reach the specified network. The IPv6 address of the next hop need not be directly connected; recursion is done to find the IPv6 address of the directly connected next hop. The address must be in the form documented in RFC 2373, specified in hexadecimal using 16-bit values between colons. • <i>interface-id</i>—Specify direct static routes from point-to-point and broadcast interfaces. With point-to-point interfaces, there is no need to specify the IPv6 address of the next hop. With broadcast interfaces, you should always specify the IPv6 address of the next hop, or ensure that the specified prefix is assigned to the link, specifying a link-local address as the next hop. You can optionally specify the IPv6 address of the next hop to which packets are sent. <p>Note You must specify an <i>interface-id</i> when using a link-local address as the next hop (the link-local next hop must also be an adjacent router).</p> <ul style="list-style-type: none"> • <i>administrative distance</i>—(Optional) An administrative distance. The range is 1 to 254; the default value is 1, which gives static routes precedence over any other type of route except connected routes. To configure a floating static route, use an administrative distance greater than that of the dynamic routing protocol.
Step 3 end	Return to privileged EXEC mode.

	Command	Purpose
Step 4	<p>show ipv6 static [<i>ipv6-address</i> <i>ipv6-prefix/prefix length</i>] [interface <i>interface-id</i>] [recursive] [detail]</p> <p>or</p> <p>show ipv6 route static [<i>updated</i>]</p>	<p>Verify your entries by displaying the contents of the IPv6 routing table.</p> <ul style="list-style-type: none"> • interface <i>interface-id</i>—(Optional) Display only those static routes with the specified interface as an egress interface. • recursive—(Optional) Display only recursive static routes. The recursive keyword is mutually exclusive with the interface keyword, but it can be used with or without the IPv6 prefix included in the command syntax. • detail—(Optional) Display this additional information: <ul style="list-style-type: none"> – For valid recursive routes, the output path set, and maximum resolution depth. – For invalid routes, the reason why the route is not valid.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

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

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

```
Switch(config)# ipv6 route 2001:0DB8::/32 gigabitethernet0/1 130
```

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

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

Displaying IPv6

Table 32-3 shows the privileged EXEC commands for monitoring IPv6 on the switch.

Table 32-3 Commands for Monitoring IPv6

Command	Purpose
show ipv6 access-list	Display a summary of access lists.
show ipv6 interface <i>interface-id</i>	Display IPv6 interface status and configuration.
show ipv6 mtu	Display IPv6 MTU per destination cache.
show ipv6 neighbors	Display IPv6 neighbor cache entries.
show ipv6 prefix-list	Display a list of IPv6 prefix lists.
show ipv6 protocols	Display IPv6 routing protocols on the switch.
show ipv6 route	Display the IPv6 route table entries.
show ipv6 static	Display IPv6 static routes.
show ipv6 traffic	Display IPv6 traffic statistics.

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

```
Switch# show ipv6 interface
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
    3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
<output truncated>
```

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

```
Switch# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip fer"
  Interfaces:
    Vlan6
    FastEthernet0/4
    FastEthernet0/11
    FastEthernet0/12
  Redistribution:
    None
```

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

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

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

```
Switch# show ipv6 neighbors
IPv6 Address                               Age Link-layer Addr State Interface
3FFE:C000:0:7::777                         - 0007.0007.0007 REACH V17
3FFE:C101:113:1::33                        - 0000.0000.0033 REACH Fa1/0/13
```

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

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

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

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

ICMP statistics:
  Rcvd: 1 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
  unreachable: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
  parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout, 0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        1 router solicit, 0 router advert, 0 redirects
        0 neighbor solicit, 0 neighbor advert
  Sent: 10112 output, 0 rate-limited
  unreachable: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
  parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout, 0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        0 router solicit, 9944 router advert, 0 redirects
        84 neighbor solicit, 84 neighbor advert

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

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

