



# CHAPTER 12

## Configuring IPv6

---

This chapter describes how to enable and configure IPv6 on the security appliance. IPv6 is available in Routed firewall mode only.

This chapter includes the following sections:

- [IPv6-enabled Commands, page 12-1](#)
- [Configuring IPv6, page 12-2](#)
- [Verifying the IPv6 Configuration, page 12-11](#)

For an sample IPv6 configuration, see [Appendix B, “Sample Configurations.”](#)

## IPv6-enabled Commands

The following security appliance commands can accept and display IPv6 addresses:

- capture
- configure
- copy
- http
- name
- object-group
- ping
- show conn
- show local-host
- show tcpstat
- ssh
- telnet
- tftp-server
- who
- write

**Note**

Failover does not support IPv6. The **ipv6 address** command does not support setting standby addresses for failover configurations. The **failover interface ip** command does not support using IPv6 addresses on the failover and Stateful Failover interfaces.

When entering IPv6 addresses in commands that support them, simply enter the IPv6 address using standard IPv6 notation, for example `ping fe80::2e0:b6ff:fe01:3b7a`. The security appliance correctly recognizes and processes the IPv6 address. However, you must enclose the IPv6 address in square brackets ([ ]) in the following situations:

- You need to specify a port number with the address, for example `[fe80::2e0:b6ff:fe01:3b7a]:8080`.
- The command uses a colon as a separator, such as the **write net** and **config net** commands, for example `configure net [fe80::2e0:b6ff:fe01:3b7a]:/tftp/config/pixconfig`.

The following commands were modified to work for IPv6:

- debug
- fragment
- ip verify
- mtu
- icmp (entered as **ipv6 icmp**)

The following inspection engines support IPv6:

- FTP
- HTTP
- ICMP
- SMTP
- TCP
- UDP

## Configuring IPv6

This section contains the following topics:

- [Configuring IPv6 on an Interface, page 12-3](#)
- [Configuring a Dual IP Stack on an Interface, page 12-4](#)
- [Enforcing the Use of Modified EUI-64 Interface IDs in IPv6 Addresses, page 12-4](#)
- [Configuring IPv6 Duplicate Address Detection, page 12-4](#)
- [Configuring IPv6 Default and Static Routes, page 12-5](#)
- [Configuring IPv6 Access Lists, page 12-6](#)
- [Configuring IPv6 Neighbor Discovery, page 12-7](#)
- [Configuring a Static IPv6 Neighbor, page 12-11](#)

## Configuring IPv6 on an Interface

At a minimum, each interface needs to be configured with an IPv6 link-local address. Additionally, you can add a site-local and global address to the interface.


**Note**

The security appliance does not support IPv6 anycast addresses.

You can configure both IPv6 and IPv4 addresses on an interface.

To configure IPv6 on an interface, perform the following steps:

**Step 1** Enter interface configuration mode for the interface on which you are configuring the IPv6 addresses:

```
hostname(config)# interface if
```

**Step 2** Configure an IPv6 address on the interface. You can assign several IPv6 addresses to an interface, such as an IPv6 link-local, site-local, and global address. However, at a minimum, you must configure a link-local address.

There are several methods for configuring IPv6 addresses. Pick the method that suits your needs from the following:

- The simplest method is to enable stateless autoconfiguration on the interface. Enabling stateless autoconfiguration on the interface configures IPv6 addresses based on prefixes received in Router Advertisement messages. A link-local address, based on the Modified EUI-64 interface ID, is automatically generated for the interface when stateless autoconfiguration is enabled. To enable stateless autoconfiguration, enter the following command:

```
hostname(config-if)# ipv6 address autoconfig
```

- If you only need to configure a link-local address on the interface and are not going to assign any other IPv6 addresses to the interface, you have the option of manually defining the link-local address or generating one based on the interface MAC address (Modified EUI-64 format):

- Enter the following command to manually specify the link-local address:

```
hostname(config-if)# ipv6 address ipv6-address link-local
```

- Enter the following command to enable IPv6 on the interface and automatically generate the link-local address using the Modified EUI-64 interface ID based on the interface MAC address:

```
hostname(config-if)# ipv6 enable
```


**Note**

You do not need to use the **ipv6 enable** command if you enter any other **ipv6 address** commands on an interface; IPv6 support is automatically enabled as soon as you assign an IPv6 address to the interface.

- Assign a site-local or global address to the interface. When you assign a site-local or global address, a link-local address is automatically created. Enter the following command to add a global or site-local address to the interface. Use the optional **eui-64** keyword to use the Modified EUI-64 interface ID in the low order 64 bits of the address.

```
hostname(config-if)# ipv6 address ipv6-address [eui-64]
```

- Step 3** (Optional) Suppress Router Advertisement messages on an interface. By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want the security appliance to supply the IPv6 prefix (for example, the outside interface).

Enter the following command to suppress Router Advertisement messages on an interface:

```
hostname(config-if)# ipv6 nd suppress-ra
```

---

## Configuring a Dual IP Stack on an Interface

The security appliance supports the configuration of both IPv6 and IPv4 on an interface. You do not need to enter any special commands to do so; simply enter the IPv4 configuration commands and IPv6 configuration commands as you normally would. Make sure you configure a default route for both IPv4 and IPv6.

## Enforcing the Use of Modified EUI-64 Interface IDs in IPv6 Addresses

RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture requires that the interface identifier portion of all unicast IPv6 addresses, except those that start with binary value 000, be 64 bits long and be constructed in Modified EUI-64 format. The security appliance can enforce this requirement for hosts attached to the local link.

To enforce the use of Modified EUI-64 format interface identifiers in IPv6 addresses on a local link, enter the following command:

```
hostname(config)# ipv6 enforce-eui64 if_name
```

The *if\_name* argument is the name of the interface, as specified by the **namif** command, on which you are enabling the address format enforcement.

When this command is enabled on an interface, the source addresses of IPv6 packets received on that interface are verified against the source MAC addresses to ensure that the interface identifiers use the Modified EUI-64 format. If the IPv6 packets do not use the Modified EUI-64 format for the interface identifier, the packets are dropped and the following system log message is generated:

```
%PIX|ASA-3-325003: EUI-64 source address check failed.
```

The address format verification is only performed when a flow is created. Packets from an existing flow are not checked. Additionally, the address verification can only be performed for hosts on the local link. Packets received from hosts behind a router will fail the address format verification, and be dropped, because their source MAC address will be the router MAC address and not the host MAC address.

## Configuring IPv6 Duplicate Address Detection

During the stateless autoconfiguration process, duplicate address detection verifies the uniqueness of new unicast IPv6 addresses before the addresses are assigned to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection is performed first on the new link-local address. When the link local address is verified as unique, then duplicate address detection is performed all the other IPv6 unicast addresses on the interface.

Duplicate address detection is suspended on interfaces that are administratively down. While an interface is administratively down, the unicast IPv6 addresses assigned to the interface are set to a pending state. An interface returning to an administratively up state restarts duplicate address detection for all of the unicast IPv6 addresses on the interface.

When a duplicate address is identified, the state of the address is set to DUPLICATE, the address is not used, and the following error message is generated:

```
%PIX|ASA-4-325002: Duplicate address ipv6_address/MAC_address on interface
```

If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface. If the duplicate address is a global address, the address is not used. However, all configuration commands associated with the duplicate address remain as configured while the state of the address is set to DUPLICATE.

If the link-local address for an interface changes, duplicate address detection is performed on the new link-local address and all of the other IPv6 address associated with the interface are regenerated (duplicate address detection is performed only on the new link-local address).

The security appliance uses neighbor solicitation messages to perform duplicate address detection. By default, the number of times an interface performs duplicate address detection is 1.

To change the number of duplicate address detection attempts, enter the following command:

```
hostname(config-if)# ipv6 nd dad attempts value
```

The *value* argument can be any value from 0 to 600. Setting the *value* argument to 0 disables duplicate address detection on the interface.

When you configure an interface to send out more than one duplicate address detection attempt, you can also use the **ipv6 nd ns-interval** command to configure the interval at which the neighbor solicitation messages are sent out. By default, they are sent out once every 1000 milliseconds.

To change the neighbor solicitation message interval, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

The *value* argument can be from 1000 to 3600000 milliseconds.

**Note**

Changing this value changes it for all neighbor solicitation messages sent out on the interface, not just those used for duplicate address detection.

## Configuring IPv6 Default and Static Routes

The security appliance automatically routes IPv6 traffic between directly connected hosts if the interfaces to which the hosts are attached are enabled for IPv6 and the IPv6 ACLs allow the traffic.

The security appliance does not support dynamic routing protocols. Therefore, to route IPv6 traffic to a non-connected host or network, you need to define a static route to the host or network or, at a minimum, a default route. Without a static or default route defined, traffic to non-connected hosts or networks generate the following error message:

```
%PIX|ASA-6-110001: No route to dest_address from source_address
```

You can add a default route and static routes using the **ipv6 route** command.

To configure an IPv6 default route and static routes, perform the following steps:

---

**Step 1** To add the default route, use the following command:

```
hostname(config)# ipv6 route if_name ::/0 next_hop_ipv6_addr
```

The address `::/0` is the IPv6 equivalent of “any.”

**Step 2** (Optional) Define IPv6 static routes. Use the following command to add an IPv6 static route to the IPv6 routing table:

```
hostname(config)# ipv6 route if_name destination next_hop_ipv6_addr [admin_distance]
```




---

**Note** The `ipv6 route` command works like the `route` command used to define IPv4 static routes.

---

## Configuring IPv6 Access Lists

Configuring an IPv6 access list is similar configuring an IPv4 access, but with IPv6 addresses.

To configure an IPv6 access list, perform the following steps:

---

**Step 1** Create an access entry. To create an access list, use the `ipv6 access-list` command to create entries for the access list. There are two main forms of this command to choose from, one for creating access list entries specifically for ICMP traffic, and one to create access list entries for all other types of IP traffic.

- To create an IPv6 access list entry specifically for ICMP traffic, enter the following command:

```
hostname(config)# ipv6 access-list id [line num] {permit | deny} icmp source
destination [icmp_type]
```

- To create an IPv6 access list entry, enter the following command:

```
hostname(config)# ipv6 access-list id [line num] {permit | deny} protocol source
[src_port] destination [dst_port]
```

The following describes the arguments for the `ipv6 access-list` command:

- *id*—The name of the access list. Use the same *id* in each command when you are entering multiple entries for an access list.
- *line num*—When adding an entry to an access list, you can specify the line number in the list where the entry should appear.
- **permit | deny**—Determines whether the specified traffic is blocked or allowed to pass.
- **icmp**—Indicates that the access list entry applies to ICMP traffic.
- *protocol*—Specifies the traffic being controlled by the access list entry. This can be the name (**ip**, **tcp**, or **udp**) or number (1-254) of an IP protocol. Alternatively, you can specify a protocol object group using **object-group** *grp\_id*.
- *source and destination*—Specifies the source or destination of the traffic. The source or destination can be an IPv6 prefix, in the format *prefix/length*, to indicate a range of addresses, the keyword **any**, to specify any address, or a specific host designated by **host** *host\_ipv6\_addr*.

- *src\_port and dst\_port*—The source and destination port (or service) argument. Enter an operator (**lt** for less than, **gt** for greater than, **eq** for equal to, **neq** for not equal to, or **range** for an inclusive range) followed by a space and a port number (or two port numbers separated by a space for the **range** keyword).
- *icmp\_type*—Specifies the ICMP message type being filtered by the access rule. The value can be a valid ICMP type number (from 0 to 155) or one of the ICMP type literals as shown in [Appendix D, “Addresses, Protocols, and Ports”](#). Alternatively, you can specify an ICMP object group using **object-group id**.

**Step 2** To apply the access list to an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface if_name
```

---

## Configuring IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMPv6 messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighboring routers.

This section contains the following topics:

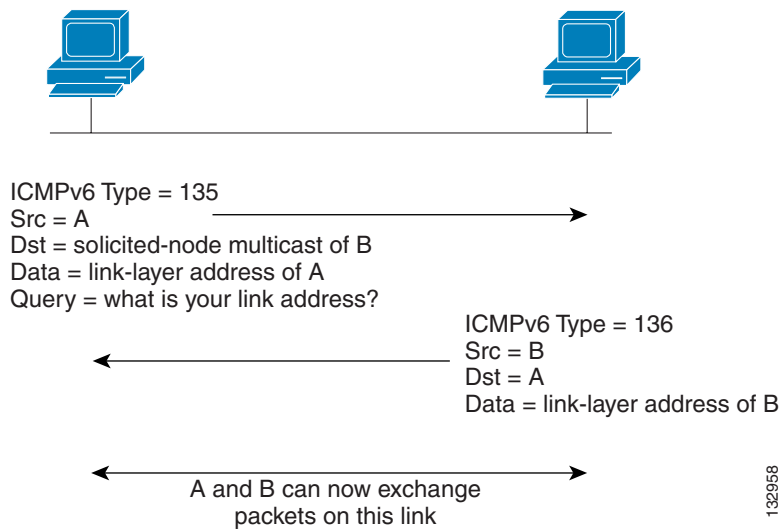
- [Configuring Neighbor Solicitation Messages, page 12-7](#)
- [Configuring Router Advertisement Messages, page 12-9](#)
- [Multicast Listener Discovery Support, page 12-11](#)

## Configuring Neighbor Solicitation Messages

Neighbor solicitation messages (ICMPv6 Type 135) are sent on the local link by nodes attempting to discover the link-layer addresses of other nodes on the local link. The neighbor solicitation message is sent to the solicited-node multicast address. The source address in the neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The neighbor solicitation message also includes the link-layer address of the source node.

After receiving a neighbor solicitation message, the destination node replies by sending a neighbor advertisement message (ICMPv6 Type 136) on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node sending the neighbor advertisement message; the destination address is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate. [Figure 12-1](#) shows the neighbor solicitation and response process.

**Figure 12-1 IPv6 Neighbor Discovery—Neighbor Solicitation Message**

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

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

You can configure the neighbor solicitation message interval and neighbor reachable time on a per-interface basis. See the following topics for more information:

- [Configuring the Neighbor Solicitation Message Interval, page 12-8](#)
- [Configuring the Neighbor Reachable Time, page 12-8](#)

### Configuring the Neighbor Solicitation Message Interval

To configure the interval between IPv6 neighbor solicitation retransmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

Valid values for the *value* argument range from 1000 to 3600000 milliseconds. The default value is 1000 milliseconds.

This setting is also sent in router advertisement messages.

### Configuring the Neighbor Reachable Time

The neighbor reachable time enables detecting unavailable neighbors. Shorter configured times enable detecting unavailable neighbors more quickly; however, shorter times consume more IPv6 network bandwidth and processing resources in all IPv6 network devices. Very short configured times are not recommended in normal IPv6 operation.

To configure the amount of time that a remote IPv6 node is considered reachable after a reachability confirmation event has occurred, enter the following command:

```
hostname(config-if)# ipv6 nd reachable-time value
```



Valid values for the *value* argument range from 0 to 3600000 milliseconds. The default is 0.

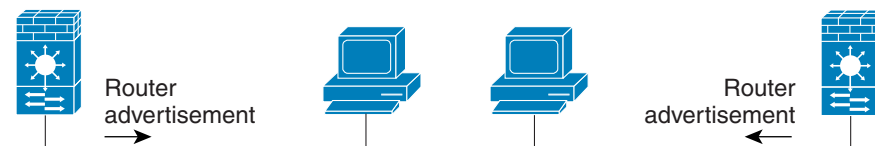
This information is also sent in router advertisement messages.

When 0 is used for the *value*, the reachable time is sent as undetermined. It is up to the receiving devices to set and track the reachable time value. To see the time used by the security appliance when this value is set to 0, use the **show ipv6 interface** command to display information about the IPv6 interface, including the ND reachable time being used.

## Configuring Router Advertisement Messages

Router advertisement messages (ICMPv6 Type 134) are periodically sent out each IPv6 configured interface of security appliance. The router advertisement messages are sent to the all-nodes multicast address.

**Figure 12-2 IPv6 Neighbor Discovery—Router Advertisement Message**



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

132917

Router advertisement messages typically include the following information:

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

Router advertisements are also sent in response to router solicitation messages (ICMPv6 Type 133). Router solicitation messages are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. Because router solicitation messages are usually sent by hosts at system startup, and the host does not have a configured unicast address, the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When a router advertisement is sent in response to a router solicitation, the destination address in the router advertisement message is the unicast address of the source of the router solicitation message.

You can configure the following settings for router advertisement messages:

- The time interval between periodic router advertisement messages.
- The router lifetime value, which indicates the amount of time IPv6 nodes should consider security appliance to be the default router.
- The IPv6 network prefixes in use on the link.
- Whether or not an interface transmits router advertisement messages.

Unless otherwise noted, the router advertisement message settings are specific to an interface and are entered in interface configuration mode. See the following topics for information about changing these settings:

- [Configuring the Router Advertisement Transmission Interval, page 12-10](#)
- [Configuring the Router Lifetime Value, page 12-10](#)
- [Configuring the IPv6 Prefix, page 12-10](#)
- [Suppressing Router Advertisement Messages, page 12-11](#)

### Configuring the Router Advertisement Transmission Interval

By default, router advertisements are sent out every 200 seconds. To change the interval between router advertisement transmissions on an interface, enter the following command:

```
ipv6 nd ra-interval [msec] value
```

Valid values range from 3 to 1800 seconds (or 500 to 1800000 milliseconds if the **msec** keyword is used).

The interval between transmissions should be less than or equal to the IPv6 router advertisement lifetime if security appliance is configured as a default router by using the **ipv6 nd ra-lifetime** command. To prevent synchronization with other IPv6 nodes, randomly adjust the actual value used to within 20 percent of the desired value.

### Configuring the Router Lifetime Value

The router lifetime value specifies how long nodes on the local link should consider security appliance as the default router on the link.

To configure the router lifetime value in IPv6 router advertisements on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ra-lifetime seconds
```

Valid values range from 0 to 9000 seconds. The default is 1800 seconds. Entering 0 indicates that security appliance should not be considered a default router on the selected interface.

### Configuring the IPv6 Prefix

Stateless autoconfiguration uses IPv6 prefixes provided in router advertisement messages to create the global unicast address from the link-local address.

To configure which IPv6 prefixes are included in IPv6 router advertisements, enter the following command:

```
hostname(config-if)# ipv6 nd prefix ipv6-prefix/prefix-length
```

**Note**

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

## Suppressing Router Advertisement Messages

By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want security appliance to supply the IPv6 prefix (for example, the outside interface).

To suppress IPv6 router advertisement transmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd suppress-ra
```

Entering this command causes the security appliance to appear as a regular IPv6 neighbor on the link and not as an IPv6 router.

## Multicast Listener Discovery Support

Multicast Listener Discovery Protocol (MLD) Version 2 is supported to discover the presence of multicast address listeners on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. ASA becomes a multicast address listener, or a host, but not a multicast router, and responds to Multicast Listener Queries and sends Multicast Listener Reports only.

The following commands were added or enhanced to support MLD:

- clear ipv6 mld traffic Command
- show ipv6 mld Command

## Configuring a Static IPv6 Neighbor

You can manually define a neighbor in the IPv6 neighbor cache. If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry. Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.

To configure a static entry in the IPv6 neighbor discovery cache, enter the following command:

```
hostname(config-if)# ipv6 neighbor ipv6_address if_name mac_address
```

The *ipv6\_address* argument is the link-local IPv6 address of the neighbor, the *if\_name* argument is the interface through which the neighbor is available, and the *mac\_address* argument is the MAC address of the neighbor interface.



### Note

The **clear ipv6 neighbors** command does not remove static entries from the IPv6 neighbor discovery cache; it only clears the dynamic entries.

## Verifying the IPv6 Configuration

This section describes how to verify your IPv6 configuration. You can use various clear, and show commands to verify your IPv6 settings.

This section includes the following topics:

- [The show ipv6 interface Command, page 12-12](#)

- [The show ipv6 route Command, page 12-12](#)
- [The show ipv6 mld traffic Command, page 12-13](#)

## The show ipv6 interface Command

To display the IPv6 interface settings, enter the following command:

```
hostname# show ipv6 interface [if_name]
```

Including the interface name, such as “outside”, displays the settings for the specified interface. Excluding the name from the command displays the setting for all interfaces that have IPv6 enabled on them. The output for the command shows the following:

- The name and status of the interface.
- The link-local and global unicast addresses.
- The multicast groups the interface belongs to.
- ICMP redirect and error message settings.
- Neighbor discovery settings.

The following is sample output from the **show ipv6 interface** command:

```
hostname# show ipv6 interface

ipv6interface is down, line protocol is down
  IPv6 is enabled, link-local address is fe80::20d:88ff:feee:6a82 [TENTATIVE]
  No global unicast address is configured
  Joined group address(es):
    ff02::1
    ff02::1:ffee:6a82
  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
```



### Note

The **show interface** command only displays the IPv4 settings for an interface. To see the IPv6 configuration on an interface, you need to use the **show ipv6 interface** command. The **show ipv6 interface** command does not display any IPv4 settings for the interface (if both types of addresses are configured on the interface).

## The show ipv6 route Command

To display the routes in the IPv6 routing table, enter the following command:

```
hostname# show ipv6 route
```

The output from the **show ipv6 route** command is similar to the IPv4 **show route** command. It displays the following information:

- The protocol that derived the route.
- The IPv6 prefix of the remote network.
- The administrative distance and metric for the route.
- The address of the next-hop router.

- The interface through which the next hop router to the specified network is reached.

The following is sample output from the **show ipv6 route** command:

```
hostname# show ipv6 route

IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
L   fe80::/10 [0/0]
    via ::, inside
L   fec0::a:0:0:a0a:a70/128 [0/0]
    via ::, inside
C   fec0:0:0:a::/64 [0/0]
    via ::, inside
L   ff00::/8 [0/0]
    via ::, inside
```

## The show ipv6 mld traffic Command

To display the MLD traffic counters in the IPv6 routing table, enter the following command:

```
hostname# show ipv6 mld traffic
```

The output from the **show ipv6 mld traffic** command displays whether the expected number of MLD protocol messages have been received and sent.

The following is sample output from the **show ipv6 mld traffic** command:

```
hostname# show ipv6 mld traffic
show ipv6 mld traffic
MLD Traffic Counters
Elapsed time since counters cleared: 00:01:19
      Received      Sent
Valid MLD Packets  1          3
Queries            1          0
Reports            0          3
Leaves             0          0
Mtrace packets    0          0
Errors:
Malformed Packets  0
Martian source    0
Non link-local source 0
Hop limit is not equal to 1 0
```

