Chapter 2
Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6) routing on your E-series router; it contains the following sections:

Overview

Internet Protocol version 6 (IPv6) is designed to eventually supersede IP version 4 (IPv4). The intent of this design change is not to take a radical step away from IPv4, but to enhance IP addressing and maintain other IPv4 functions that work well.

The differences between IPv4 and IPv6 include the following:

- Expanded addressing capabilities

  IPv6 increases the size of the IP address from 32 bits to 128 bits. This increased size provides a larger address space and a much larger number of addressable nodes.

- Simplified header format

  Reducing some common processing costs associated with packet handling and streamlining the bandwidth cost of the larger IPv6 header, some IPv4-specific header fields either no longer exist or are now optional in the IPv6 header.

- Traffic flow labelling capabilities

  The ability to label packets for specific traffic flows exists in the IPv6 packet. These labels allow for nondefault quality of service (QoS) or the possibility of "real-time" services.

- Authentication capabilities

  Authentication provides the ability to use extensions for some authentication and data integrity applications.

IPv6 continues to provide the basic packet delivery service for all TCP/IP networks. As a *connectionless* protocol, IPv6 does not exchange control information to establish an end-to-end connection before transmitting data. Instead, just like its IPv4 predecessor, IPv6 continues to rely on protocols in other layers to establish the connection if connection-oriented services are required and to provide error detection and error recovery.

In addition to supporting a revised header structure and an expanded addressing format, the E-series router supports the following IPv6 features:

- Static routes
- ICMPv6
- Ping
- Traceroute
- Routing policy (See *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy* for details.)
- IPv6 B-RAS (See the *JUNOSe Broadband Access Configuration Guide* for details.)
- IPv6 tunnel routing tables

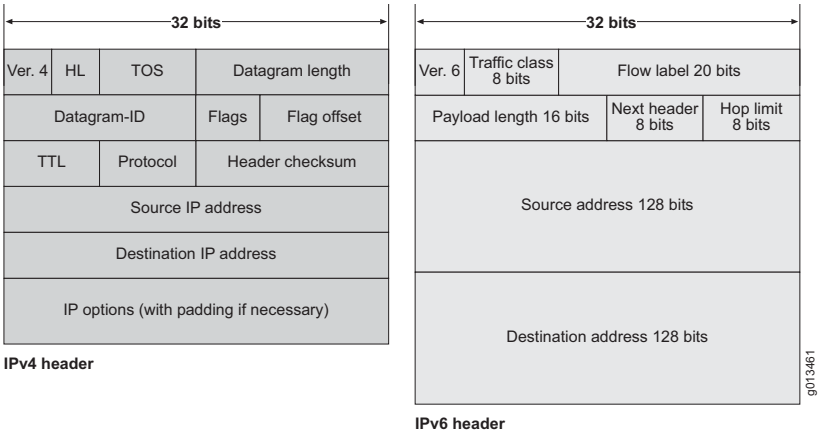
IPv6 Packet Headers

An IPv6 packet is a block of data that contains a header and a payload. The header is the information necessary to deliver the packet to a destination address; the payload is the data that you want to deliver. IPv6 packets can use a standard or an extended format.

IPv4 and IPv6 Header Differences

The main difference between IPv4 and IPv6 resides in their headers. Figure 13 provides a comparison between the two protocol versions.

Figure 13: IPv4 and IPv6 Header Comparison



Standard IPv6 Headers

IPv6 packet headers contain many of the fields found in IPv4 packet headers; some of these fields differ from IPv4. (See Figure 13.)

The 40-byte IPv6 header consists of the following eight fields:

- Version—Indicates the version of the Internet Protocol.
- Traffic class—Previously the type-of-service (ToS) field in IPv4, the traffic class field defines the class-of-service (CoS) priority of the packet. However, the semantics for this field (for example, DiffServ code points) are identical to IPv4.
- Flow label—The flow label identifies all packets belonging to a specific flow (that is, packet flows requiring a specific class of service [CoS]); routers can identify these packets and handle them in a similar fashion.
- Payload length—Previously the total length field in IPv4, the payload length field specifies the length of the IPv6 payload.
- Next header—Previously the protocol field in IPv4, the Next Header field indicates the next extension header to examine.

- Hop limit—Previously the time-to-live (TTL) field in IPv4, the hop limit indicates the maximum number of hops allowed.
- Source address—Identifies the address of the source node sending the packet.
- Destination address—Identifies the final destination node address for the packet.

Extension Headers

In IPv6, extension headers are used to encode optional Internet-layer information. Extension headers are placed between the IPv6 header and the upper-layer header in a packet.

IPv6 enables you to chain extension headers together by using the next header field. The next header field, located in the IPv6 header, indicates to the router which extension header to expect next. If there are no more extension headers, the next header field indicates the upper-layer header (TCP header, UDP header, ICMPv6 header, an encapsulated IP packet, or other items).

IPv6 Addressing

IPv6 increases the size of the IP address from the 32 bits found in IPv4 to 128 bits. This increased size provides for a broader range of addressing hierarchies and a much larger number of addressable nodes.

In addition to the increased size, IPv6 addresses can be of different scopes that categorize what types of applications are suitable for the address. IPv6 does not support broadcast addresses, but uses multicast addresses to serve this role. In addition, IPv6 also defines a new type of address called *anycast*.

Address Representation

IPv6 addresses consist of eight hexadecimal groups. Each hexadecimal group, separated by a colon (:), consists of a 16-bit hexadecimal value. The following is an example of the IPv6 format:

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

A group of *xxxx* represents the 16-bit hexadecimal value. Each individual *x* represents a 4-bit hexadecimal value. The following is an example of a possible IPv6 address:

4FDE:0000:0000:0002:0022:F376:FF3B:AB3F

NOTE: Hexadecimal letters in IPv6 addresses are not case sensitive.

IPv6 Address Compression

IPv6 addresses often contain consecutive hexadecimal fields of zeros. To simplify address entry, you can use two colons (::) to represent the consecutive fields of zeros when typing the IPv6 address. Table 9 provides compressed IPv6 address format examples.

Table 9: Compressed IPv6 Formats

IPv6 Address Type	Full Format	Compressed Format
Unicast	10FB:0:0:0:C:ABC:1F0C:44DA	10FB::C:ABC:1F0C:44DA
Multicast	FD01:0:0:0:0:0:0:1F	FD01::1F
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

NOTE: You can use two colons (::) only once in an IPv6 address to represent hexadecimal fields of consecutive zeros.

IPv6 Address Prefix

An IPv6 address prefix is a combination of an IPv6 prefix (address) and a prefix length. The prefix takes the form *ipv6-prefix/prefix-length* and represents a block of address space (or a network). The *ipv6-prefix* variable follows general IPv6 addressing rules (see RFC 2373 for details). The */prefix-length* variable is a decimal value that indicates the number of contiguous, higher-order bits of the address that make up the network portion of the address. For example, 10FA:6604:8136:6502::/64 is a possible IPv6 prefix.

Address Types

IPv6 can use several types of addresses:

- Unicast—Used to identify a single interface, this release of the E-series router product supports the following unicast address types:
  - Global aggregatable—Provides for aggregation of routing prefixes to limit the number of global routing table entries
  - Link-local—Eliminates the need for a globally unique prefix. Local-link addresses allow communications between devices on a local link.
  - Site-local—Used as private addresses to restrict communication to a domain portion.

NOTE: IPv6 routers must not forward packets that have site-local source or destination addresses outside the site.

  - IPv4-compatible—Contains a standard IPv4 address in the lower-order 32 bits of the address and zeros in the higher-order 96 bits of the address. For example, the format of an IPv4-compatible IPv6 address is 0:0:0:0:0:0:A.B.C.D (or condensed as ::A.B.C.D). In other words, devices using IPv6 use the entire 128-bit IPv4-compatible IPv6 address, whereas IPv4 devices use the IPv4 address embedded within the lower-order 32-bits of the address. You would use IPv4-compatible IPv6 addresses for devices that must support both IPv4 and IPv6 protocols.

- Multicast—Used for sending packets to multiple destinations. A multicast transmission sends packets to all interfaces that are part of a multicast group. The group is represented by the IPv6 destination address of the packet.
- Anycast – Used for a set of interfaces on different nodes. An anycast transmission sends packets to only one of the interfaces associated with the address, not to all of the interfaces. This interface is typically the closest interface, as defined by the routing protocol.
- Loopback—Used by a node to send an IPv6 packet to itself. An IPv6 loopback address functions the same as an IPv4 loopback address.
- Unspecified—Indicates the absence of an IPv6 address. For example, newly initialized IPv6 nodes may use the unspecified address as the source address in their packets until they receive an IPv6 address.

NOTE: IPv6 does not use broadcast addresses; instead, IPv6 uses multicast addresses.

Address Scope

Some unicast and multicast IPv6 addresses contain a value known as *scope*. This value identifies the application suitable for the address.

Unicast addresses support two types of scope—global and local. In addition, there are two types of local scope—link-local addresses and site-local addresses.

Link-local unicast addresses, identified by the first ten bits of the prefix, function within a single network link. You cannot use link-local addresses outside a network link.

Site-local unicast addresses function within a site or an intranet. A site consists of multiple network links, and site-local addresses identify nodes inside the intranet. You cannot use site-local addresses outside the site.

Multicast addresses support 16 different types of scope, including node, link, site, organization, and global scope. A four-bit field in the prefix identifies the scope.

Address Structure

Unicast addresses identify a single interface. The address consists of *n* bits for the prefix and 128-*n* bits for the interface ID.

Multicast addresses identify a set of interfaces. The address is made up of the first 8 bits of all ones, a 4-bit flag field, a 4-bit scope field, and a 112-bit group ID.

11111111 | *flgs* | *scop* | *group ID*

The first octet of ones identifies the address as a multicast address. The flags field identifies whether the multicast address is a well-known address or whether it is a transient multicast address. The scope field identifies the scope of the multicast address. The 112-bit group ID identifies the multicast group.

Similar to multicast addresses, anycast addresses identify a set of interfaces. However, packets are sent to only one of the interfaces, not to all interfaces. Anycast addresses are allocated from the normal unicast address space and cannot be distinguished from a unicast address in format. Therefore, each member of an anycast group must be configured to recognize certain addresses as anycast addresses.

ICMP Support

Internet Control Message Protocol (ICMP) provides a mechanism that enables a router or destination host to report an error in data traffic processing to the original source of the packet. For this release, the E-series router supports ICMP for use in the IPv6 **ping** and **traceroute** commands.

The **ping** and **traceroute** commands help you determine destination reachability within a network.

- Use the **ping ipv6** command to send an ICMP echo request packet. In the following example, the request packet is sent to address 1::1 with a data size of 200 and a timeout value of 10 seconds:

  host1#**ping ipv6 1::1 data-size 200 timeout 10**

- Use the **traceroute ipv6** command to discover routes that router packets follow when traveling to their destination. In the following example, the trace destination address is 1::1, the maximum number of hops of the trace is 20, and the timeout value is 10 seconds:

  host1#**traceroute ipv6 1::1 hop-limit 20 timeout 10**

IPv6 Tunnel Routing Table

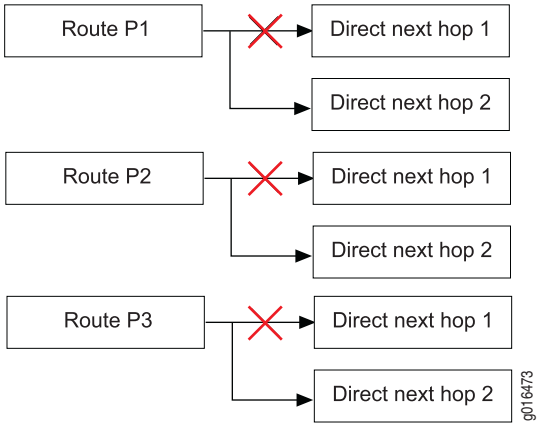
The IPv6 tunnel routing tables include IPv6 routes that point only to tunnels, such as MPLS tunnels. The tunnel routing table is not used for forwarding. Instead, protocols resolve next hops by looking up the routes that point to tunnels. The routes in the tunnel routing table cannot be redistributed. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS* for more information.

Indirect Next Hop Support

The router uses indirect next hops to promote faster network convergence (for example, in BGP networks) by decreasing the number of routing table changes required when a change in the network topology occurs.

Direct next-hops point routes in the routing table toward individual, direct next-hop connections. (See Figure 14.)

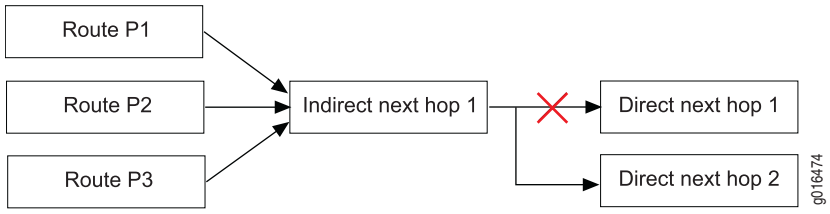
Figure 14: Direct Next Hops



Indirect next hops enable multiple routes in the routing table to point to a single next hop, thereby accelerating convergence. (See Figure 15.)

NOTE: Indirect next hops are not limited to any number of levels. In other words, an indirect next hop can point to a direct next hop or another indirect next hop.

Figure 15: Indirect Next Hops



By using indirect next hops, if a topology change occurs in the network, only the indirect next hop is modified in the routing table, decreasing the number of state changes required to achieve convergence.

Platform Considerations

For information about modules that support IPv6 and Neighbor Discovery on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IP.

For information about modules that support IP on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support IP.

References

For more information about IPv6, consult the following resources:

- RFC 2373—IP Version 6 Addressing Architecture (July 1998)
- RFC 2460—Internet Protocol, Version 6 (IPv6) (December 1998)
- RFC 2461—Neighbor Discovery for IP Version 6 (IPv6) (December 1998)
- RFC 2462—IPv6 Stateless Address Autoconfiguration (December 1998)
- RFC 2463—Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (December 1998)
- RFC 2464—Transmission of IPv6 Packets over Ethernet Networks (December 1998)
- RFC 2465—Management Information Base for IP Version 6: Textual Conventions and General Group (December 1998)
- RFC 2466—Management Information Base for IP Version 6: ICMPv6 Group (December 1998)

You can access these and other Internet RFCs and drafts at the following URL:

http://www.ietf.org

Before You Configure IPv6

Before you configure IPv6, you must create the lower-layer interfaces over which IPv6 traffic flows. In this release, the following modules support IPv6 configuration:

- ATM OC3/STM-1
- ATM OC12/STM-4
- Fast Ethernet (FE-8)
- Gigabit Ethernet (GE)
- 10-Gigabit Ethernet (10GE)
- OC48 POS (PPP only)

For example, to configure an ATM interface:

```
host1(config)#interface atm 1/0
host1(config-if)#atm sonet stm-1
host1(config-if)#no loopback
host1(config-if)#atm clock internal chassis
host1(config-if)#interface atm 1/0.10
host1(config-if)#atm pvc 10 0 20 aal5snap
```

See *JUNOSe Link Layer Configuration Guide, Chapter 1, Configuring ATM* for information about configuring an ATM interface. See *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces* for information about configuring an Ethernet interface.

Configuring an IPv6 License

You must configure an IPv6 license before you can use any IPv6 commands on the E-series router.

license ipv6

- Use to specify an IPv6 license.
- Purchase an IPv6 license to allow IPv6 configuration on the E-series router.

NOTE: Acquire the license from Juniper Networks Customer Services and Support or your Juniper Networks sales representative.

- Example

  host1(config)#**license ipv6** *license-value*

- Use the **no** version to disable the license.

Creating an IPv6 Profile

You can configure an IPv6 interface dynamically by creating a profile. A profile is a set of characteristics that acts as a pattern that can be dynamically assigned to an IPv6 interface. You can manage a large number of IPv6 interfaces efficiently by creating a profile with a specific set of characteristics. In addition, you can create a profile to assign an IPv6 interface to a virtual router.

A profile can contain one or more of the following characteristics:

- address—Configures an IPv6 address on an interface
- mld—Configures the MLD interface
- mtu—Configures the MTU for a network
- nd—Configures Neighbor Discovery (ND) router advertisement characteristics
- policy—Attaches (or removes) a policy to (or from) an interface
- sa-validate—Enables source address validation
- unnumbered—Configures IPv6 on this interface without a specific address
- virtual-router—Specifies a virtual router to which interfaces created by this profile will be attached

NOTE: You can also configure any of these IPv6 characteristics outside the profile configuration mode.

Use the **profile** command from Global Configuration mode to create or edit a profile. See *JUNOSe Link Layer Configuration Guide, Chapter 15, Configuring Dynamic Interfaces* for information about creating profiles and on other characteristics that can be applied to the profile.

```
host1(config)#profile boston
host1(config-profile)#ipv6 virtual-router warf
host1(config-profile)#ipv6 unnumbered atm 3/0
```

ipv6 address

- Use to add an IPv6 address to an interface or a subinterface.
- Example

```
host1(config)#interface atm 1/0.25
host1(config-if)#ipv6 address 1::1/64
```

NOTE: You can use this command in Interface Configuration or Subinterface Configuration mode.

- Use the **no** version of this command to remove an IPv6 address.

ipv6 nd

- Use to enable the IPv6 Neighbor Discovery process on an interface.
- You can include the following commands in IPv6 profiles to configure Neighbor Discovery route advertisement characteristics. For additional information, see *Using IPv6 Profiles and RADIUS to Configure Neighbor Discovery Route Advertisements* in *Chapter 3, Configuring Neighbor Discovery*.

Command	Description
ipv6 nd	Enables Neighbor Discovery on an interface
ipv6 nd managed-config-flag	Sets the "managed address configuration" flag in IPv6 router advertisements
ipv6 nd other-config-flag	Sets the "other stateful configuration" flag in IPv6 router advertisements
ipv6 nd prefix-advertisement	Specifies IPv6 prefix included in IPv6 router advertisements
ipv6 nd ra-interval	Configures the interval between IPv6 router advertisements
ipv6 nd ra-lifetime	Configures the router advertisement lifetime
ipv6 nd reachable-time	Configures the amount of time the router can reach an IPv6 node after a reachability confirmation event occurs
ipv6 nd suppress-ra	Disables router advertisement transmissions

- Example

  host1(config)#**profile ProfileIpv6South22**
  host1(config-profile)#**ipv6 nd**

- Use the **no** version to disable the Neighbor Discovery process for the profile.

ipv6 mtu

- Use to set the MTU size of IPv6 packets sent on an interface.
- The range is 128–10240.
- Example

  host1(config-if)#**ipv6 mtu 1000**

- Use the **no** version to restore the default MTU size.

ipv6 unnumbered

- Use to set up an unnumbered interface.
- An unnumbered interface does not have an IPv6 address assigned to it. Unnumbered interfaces are often used in point-to-point connections where an IPv6 address is not required.
- This command enables IPv6 processing on an interface without your having to assign an explicit IPv6 address to the interface.

- You supply an interface location that is the type and number of another interface on which the router has an assigned IPv6 address. This interface cannot be another unnumbered interface.
- Example

  host1(config-if)#**ipv6 unnumbered loopback 0**

- Use the **no** version to disable IPv6 processing on an interface.

ipv6 virtual-router

- Use to assign a virtual router to a profile.
- You can configure a virtual router using RADIUS instead of adding one to the profile by using the **ipv6 virtual-router** command.
- Example

  host1(config-profile)#**ipv6 virtual-router VR6**

- Use the **no** version to remove the virtual router assignment.

Assigning a Profile

To assign a profile to an interface, use the **profile** command from Interface mode.

profile

- Use to assign a profile to a PPP interface. The profile configuration is used to dynamically create an upper IP interface.
- Example

  host1(config-if)#**interface atm 3/1.50**
  host1(config-if)#**encapsulation ppp**
  host1(config-if)#**profile boston**

- Use the **no** version to remove the assignment from the interface.

Enabling Source Address Validation

Source address validation verifies that a packet has been sent from a valid source address. When a packet arrives on an interface, the router performs a routing table lookup using the source address. The result from the routing table lookup is an interface to which packets destined for that address are routed. This interface must match the interface on which the packet arrived. If it does not match, the router drops the packet.

CAUTION: When the routing table lookup for a source address contains an ECMP route, the router returns a list of interfaces for multiple next-hops. One of the interfaces in this list must match the interface on which the packet arrived or the router drops the packet. If the ECMP route uses indirect next-hops, the returned list of interfaces does not include interfaces that are reachable by those indirect next-hops. For example, if a packet arrives on an interface with source address validation enabled, and the interface is represented only by an indirect next-hop, a match for that interface does not appear in the list of interfaces from the routing table lookup. The router drops the packet.

ipv6 sa-validate

- Use to enable source address validation. Source address validation verifies that a packet has been sent from a valid source address.
- Example

  host1(config-if)#**ipv6 sa-validate**

- Use the **no** version to disable source address validation.

Establishing a Static Route

You can set a destination to receive and send traffic by a specific route through the network.

ipv6 route

- Use to establish a static IPv6 route.
- You can set a destination to receive and send traffic from and to a network or to use a specific route through the network.
- Example

  host1(config)#**ipv6 route 7fff::0/16 1::1**

- Use the **no** version of this command to remove a static route from the routing table.

Specifying an IPv6 Hop Count Limit

You can specify the maximum number of hops that the router can use in router advertisements and all IPv6 packets.

ipv6 hop-limit

- Use to set the maximum number of hops that the router can use in router advertisements and all IPv6 packets.
- Example

  host1(config)#**ipv6 hop-limit 50**

- Use the **no** version to set the hop limit for IPv6 packets to 255 hops and router advertisements to zero (0) hops (or "unspecified").

Managing IPv6 Interfaces

You can manage IPv6 interfaces in the following ways:

- Disable or reenable an IPv6 interface.

  host1(config-if)#**no ipv6 enable**
  host1(config-if)#**ipv6 enable**

- Set a baseline for IPv6 interface counters.

  host1#**clear ipv6 interface atm 2/0**

- Determine reachability within a network.

  host1#**ping ipv6 1::1**
  host1#**traceroute ipv6 1::1**

clear ipv6 interface

- Use to set a baseline for counters on a specified IPv6 interface.
- Example

  host1#**clear ipv6 interface atm 2/0**

- There is no **no** version.

ipv6 enable

- Use to enable or disable an IPv6 interface at any time.

NOTE: By default, an IPv6 interface is enabled when you first create it.

- Example

  host1(config-if)#**ipv6 enable**

- Use the **no** version of this command to disable IPv6 on an interface or a subinterface.

ping ipv6

- Use to send an ICMP echo request packet to the IPv6 address that you specify.
- Use the **source interface** keywords to specify a source interface other than the one from which the probe originates.
- Use the **source address** keywords to specify a source IP address other than the one from which the probe originates.
- You can specify the following options:
  - **packetCount**—Number of packets to send to the destination IPv6 address. If you specify a zero (0), echo requests packets are sent indefinitely.
  - **data-pattern**—Sets the type of bits contained in the packet to all ones, all zeros, a random mixture of ones and zeros, or a specific hexadecimal data pattern that can range from 0x0–0xFFFFFFFF. The default is all zeros.
  - **data-size**—Sets the number of bytes comprising the IPv6 packet and reflected in the IPv6 header in the range 0–64000; the default is 100 bytes
  - **extended** header attributes—Set the interface type and specifier of a destination address on the router that is configured for external loopback; the command succeeds only if the specified interface is configured for external loopback
  - **sweep-interval**—Specifies the change in the size of subsequent ping packets while sweeping across a range of sizes. For example, you can configure the sweep interval to sweep across the range of packets from 100 bytes to 1000 bytes in increments equal to the sweep interval. By default the router increments packets by one byte; for example, it sends 100, 101, 102, 103, ... 1000. If the sweep interval is 5, the router sends 100, 105, 110, 115, ... 1000.
  - **sweep-sizes**—Enables you to vary the sizes of the echo packets being sent. This capability is useful for determining the minimum sizes of the MTUs configured on the nodes along the path to the destination address. This reduces packet fragmentation, which contributes to performance problems. The default is not to sweep (all packets are the same size).
  - **timeout**—Sets the number of seconds to wait for an ICMP echo reply packet before the connection attempt times out
  - **hop-limit**—Sets the time-to-live hop count in the range 1–255; the default is 255

- The following characters can appear in the display after you issue the **ping** command:
  - !—Reply received
  - .—Timed out while waiting for a reply
  - ?—Unknown packet type
  - A—Admin unreachable
  - b—Packet too big
  - H—Host unreachable
  - N—Network unreachable
  - P—Port unreachable
  - p—Parameter problem
  - S—Source beyond scope
  - t—Hop limit expired (TTL expired)
- Example

  host1#**ping ipv6 1::1**

- There is no **no** version.

traceroute ipv6

- Use to discover the routes that router packets follow when traveling to their destination.
- You can specify:
  - Destination IPv6 address
  - Source interface for each of the transmitted packets
  - Source IPv6 address for each of the transmitted packets
  - Maximum number of hops of the trace and a timeout value
  - Size of the IPv6 packets (not the ICMP payload) in the range 0–64000 bytes sent with the **traceroute** command. Including a size might help locate any MTU problems that exist between your router and a particular device.
  - Hop count in the range 1–255; the default is 32
- You can also force transmission of the packets on a specified interface regardless of what the IPv6 address lookup indicates.
- Example

  host1#**traceroute ipv6 1::1 timeout 10**

- There is no **no** version.

Configuring Shared IPv6 Interfaces

You can create multiple *shared* IPv6 interfaces over the same layer 2 logical interface—for example, atm 5/3.101—enabling more than one IPv6 interface to share the same logical resources.

For additional information about shared interfaces, see *Shared IP Interfaces* on page 55.

To share IPv6 interfaces:

1. Create a layer 2 interface.

   host1(config)#**interface atm 5/3**
   host1(config-if)#**interface atm 5/3.101**

2. (Optional) Create a primary IPv6 interface.

   host1(config-if)#**ipv6 address 1::1/64**
   host1(config-if)#**exit**

3. Create the shared IPv6 interface.

   host1(config)#**interface ipv6 si0**

4. Associate the shared IPv6 interface with the layer 2 interface by the following method:

   host1(config-if)#**ipv6 share-interface atm 5/3.101**

5. To fully configure the shared interface, assign an address (or make the interface unnumbered).

   host1(config-if)#**ipv6 address 1::1/64**

interface ipv6

- Use to create an IPv6 interface for interface sharing.
- Use the specified name to refer to the shared IPv6 interface; you cannot use the layer 2 interface to refer to them, because the shared interface can be moved.
- Example

  host1(config)#**interface ipv6 si1**

- Use the **no** version to delete the IPv6 interface.

ipv6 share-interface

- Use to specify the layer 2 interface used by a shared IPv6 interface. The command fails if the layer 2 interface does not yet exist. The command is not supported (that is, it fails) if you use an RSVP tunnel (for example, **tunnel mpls:1**) to identify the layer 2 interface.
- After creating the shared IPv6 interface, you can configure it as you do any other IPv6 interface.
- The shared interface is operationally up when the layer 2 interface is operationally up.
- You can create operational shared IPv6 interfaces in the absence of a primary IPv6 interface.
- Example

  host1(config-if)#**ipv6 share-interface atm 5/3.101**

- Use the **no** version to remove the association between the layer 2 interface and the shared IPv6 interface. You can delete shared and primary IPv6 interfaces independently.

Adding a Description

The router enables you to add a text description or an alias to an IPv6 interface or subinterface. Adding a description helps you identify the interface and keep track of interface connections.

ipv6 description

- Use to assign a text description or an alias to an IPv6 interface or subinterface.
- The description or alias can be a maximum of 256 characters.
- Use the **show ipv6 interface** command to display the text description.
- Example 1

  host1(config-if)#**ipv6 description boston01 ipv6 interface**

- Example 2

  host1(config-subif)#**ipv6 description dallas05 ipv6 subinterface**

- Use the **no** version to remove the text description or alias.

IPv6 TCP Configuration

IPv6 supports TCP configuration. You use the same commands to configure TCP on IPv6 as you do to configure TCP on IPv4.

Setting MSS for TCP Connections

MSS is used by TCP to define the maximum amount of data that a TCP interface can accept in any single packet (or segment size). The MSS value is typically negotiated during connection establishment and is not renegotiated.

By default, the router uses an MSS value of 1280 bytes and the advertised MSS is derived from the MTU of the transmitting interface. However, you can use the **tcp mss** command to set the MSS for TCP use.

tcp mss

- Use to specify the MSS value for TCP to use.

NOTE: The MSS value is equal to the MTU value minus the IPv6 and TCP headers, so the MSS value is generally 60 bytes less than the MTU.

- Use the *vrfName* variable to specify a VRF to which you want to assign the TCP MSS value.
- Example

  host1(config)#**tcp mss 1000**

- Use the **no** version to remove the MSS value so that the router uses the advertised MSS derived from the MTU of the output interface.

Configuring Path MTU Discovery

IPv6 hosts transmit large amounts of data to other hosts using a series of IPv6 datagrams. To best use resources, increase performance, and avoid difficult reassembly, hosts try to send datagrams that are as large as possible without requiring fragmentation anywhere along the path from the source to the destination. This datagram size is referred to as the *path MTU* (*PMTU*), and it is equal to the smallest MTU for each hop in the path.

Path MTU discovery is the process of discovering the PMTU value and using that value when transmitting IP datagrams.

Enabling PMTU Discovery

Use the **tcp path-mtu-discovery** command to enable PMTU discovery on the active virtual router.

tcp path-mtu-discovery

- Use to enable and configure path MTU discovery on the virtual router.
- Issue the command without any keywords to enable path MTU discovery.
- Issue the **age-timer** keyword to set the time (*minutes*) that TCP waits before attempting to increase the path MTU after receiving an ICMP Too Big message or after previously increasing the PMTU successfully (*minutes2*). The range of these two timers is 1–30 minutes. The timer defaults to 10 minutes.
- Issue the **age-timer indefinite** keyword to disable PMTU aging functions.
- Example 1—Enables path MTU discovery

  host1:VR1(config)#**tcp path-mtu-discovery**

- Example 2—Sets path MTU discovery age timers differently

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 20 15**

- Example 3—Sets path MTU discovery age timers to the same value (5 minutes)

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 5**

- Example 4—Disables path MTU discovery age timers

  host1:VR1(config)#**tcp path-mtu-discovery age-timer indefinite**

- Use the **no** version with a keyword to return the values to their defaults.
- Issue the **no** version without any keywords to disable path MTU discovery on the virtual router.

Limiting PMTU

You can limit calculated PMTU values within a range by using the **tcp path-mtu-discovery max-mtu** and **tcp path-mtu-discovery min-mtu** commands. When specifying PMTU limits, keep the following in mind:

- If a PMTU discovery value is lower than the configured minimum MTU setting, PMTU discovery is disabled for that connection.
- If a PMTU discovery value is larger than the configured maximum MTU setting, the configured maximum MTU setting is used.
- The maximum MTU setting must be greater than the minimum MTU setting.

tcp path-mtu-discovery max-mtu

- Use to limit the maximum MTU size used for the path MTU.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery max-mtu 512**

- Use the **no** version to remove any limitation so that the virtual router uses the path MTU discovery value.

tcp path-mtu-discovery min-mtu

- Use to specify the minimum MTU value used for the path MTU. If the discovered PMTU value is less than the minimum setting, path MTU discovery is disabled for this connection.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery min-mtu 255**

- Use the **no** version to remove any limitation so that the virtual router uses the discovered path MTU value.

Specifying Black Hole Thresholds

Some domains might be configured not to generate certain ICMP messages (like an ICMP destination unreachable message) or to filter all ICMP messages. Under these conditions, the source of oversized ICMP packets never learns that it is sending oversized packets. The device continues sending oversized packets that never get through. This behavior is often referred to as a *black hole*.

A black hole threshold is a limit to the number of times a virtual router can retransmit identical sequences of datagrams before the retransmissions are identified as a problem.

tcp path-mtu-discovery black-hole-detect-threshold

- Use to specify the number of permitted retransmissions before the retransmissions are determined to be a problem.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery black-hole-detect-threshold 200**

- Use the **no** version to disable black hole threshold detection.

Protecting Against TCP RST or SYN DoS Attacks

You can use the **tcp ack-rst-and-syn** command to help protect the router from denial of service (DoS) attacks.

Normally, when it receives an RST or SYN message for an existing connection, TCP attempts to shut down the TCP connection. This action is expected under normal conditions, but someone maliciously generating otherwise valid RST or SYN messages can cause problems for network applications and the network as a whole.

When you enable the **tcp ack-rst-and-syn** command, the router challenges any RST or SYN messages that it receives by sending an ACK message back to the expected source of the message. The source reacts in one of the following ways:

- If the source did send the RST or SYN message, it recognizes the ACK message to be spurious and resends another RST or SYN message. The second RST or SYN message causes the router to shut down the connection.
- If the source did not send the RST or SYN message, the source accepts the ACK message as part of an existing connection. As a result, the source does not send another RST or SYN message and the router does not shut down the connection.

NOTE: Enabling this command slightly modifies the way TCP processes RST or SYN messages to ensure that they are genuine.

tcp ack-rst-and-syn

- Use to help protect the router from TCP RST and SYN denial of service attacks.
- Example

  host1(config)#**tcp ack-rst-and-syn**

- Use the **no** version to disable this protection (the default mode).

Preventing TCP PAWS Timestamp DoS Attacks

The TCP Protect Against Wrapped Sequence (PAWS) number option works by including the TCP timestamp option in all TCP headers to help validate the packet sequence number.

Normally, in PAWS packets that have the timestamps option enabled, hosts use an internal timer to compare the value of the timestamp associated with incoming segments against the last valid timestamp the host recorded. If the segment timestamp is larger than the value of the last valid timestamp, and the sequence number is less than the last acknowledgement sent, the host updates its internal timer with the new timestamp and passes the segment on for further processing.

If the host detects a segment timestamp that is smaller than the value of the last valid timestamp or the sequence number is greater than the last acknowledgement sent, the host rejects the segment.

A remote attacker can potentially determine the source and destination ports and IP addresses of both hosts that are engaged in an active connection. With this information, the attacker might be able to inject a specially crafted segment into the connection that contains a fabricated timestamp value. When the host receives this fabricated timestamp, it changes its internal timer value to match. If this timestamp value is larger than subsequent timestamp values from valid incoming segments, the host determines the incoming segments as being too old and discards them. The flow of data between hosts eventually stops, resulting in a denial of service condition.

Use the **tcp paws-disable** command to disable PAWS processing.

NOTE: Disabling PAWS does not disable other processing related to the TCP timestamp option. This means that even though you disable PAWS, a fabricated timestamp that already exists in the network can still pollute the database and result in a successful DoS attack. Enabling PAWS resets the saved timestamp state for all connections in the virtual router and stops any existing attack.

tcp paws-disable

- Use to disable the Protect Against Wrapped Sequence (PAWS) number option in TCP segments.
- You can specify a VRF context for which you want PAWS disabled.
- Example

  host1(config)#**tcp paws-disable**

- Use the **no** version to restore PAWS processing (the default mode).

Protecting Against TCP Out of Order DoS Attacks

You can use the group of **tcp resequence-buffers** commands to help protect the router from TCP out-of-order packet DoS attacks.

TCP guarantees that applications receive data in order. This means that TCP buffers any out-of-order packets it receives until ordered delivery can occur.

To prevent connections from consuming too many resources, TCP limits the amount of data it accepts to the number of data bytes that the receiver is willing to receive and buffer. TCP does not take into account the buffering scheme that the receiver uses. If the receiver uses a fixed-size receive buffer (that is, buffering all packets) regardless of length, a packet that contains only one data byte might consume many data bytes of buffer space, but only one byte of TCP space.

Under these conditions, an attacker can send a large number of 1-byte packets to an E-series router in which each packet is buffered, consuming an entire packet buffer and eventually consuming a large amount of resources.

To defend against this sort of attack, you can set defaults and limits on the number of outstanding buffers on reordering queues. You can configure these defaults and limits on a per-router, per-virtual router, or per-connection within the virtual router basis.

Limiting Buffers per Router

The **tcp resequence-buffers global-maximum** command enables you to limit the number of outstanding buffers on the entire router.

tcp resequence-buffers global-maximum

- Use to specify a router-wide maximum number of buffers that resequencing queues can contain.
- Specify a value of zero (0) to turn off the limit.
- Example

  host1(config)#**tcp resequence-buffers global-maximum**

- Use the **no** version to revert the global maximum buffer value to its default, 1000 buffers.

Limiting Buffers per Virtual Router

The **tcp resequence-buffers vr-maximum** command and **tcp resequence-buffers default-vr-maximum** command allow you to limit the number of outstanding buffers on existing or newly established virtual routers.

tcp resequence-buffers default-vr-maximum

- Use to specify the default buffer limit assigned to all virtual routers when the virtual router is established.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers default-vr-maximum 200**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

tcp resequence-buffers vr-maximum

- Use to define the maximum number of buffers that the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers vr-maximum**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

Limiting Buffers per Connection

The **tcp resequence-buffers connection-maximum** command and **tcp resequence-buffers default-connection-maximum** command allow you to limit the number of outstanding buffers on existing or newly established connections.

tcp resequence-buffers connection-maximum

- Use to define the maximum number of buffers that connections on the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the connection maximum.
- Example

  host1(config)#**tcp resequence-buffers connection-maximum 50**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

tcp resequence-buffers default-connection-maximum

- Use to specify the default buffer limit assigned to all TCP connections on a virtual router unless a specific limit is set for the VR in which the connection is established.
- Specify a value of zero (0) buffers to turn off the default limit.
- Example

  host1(config)#**tcp resequence-buffers default-connection-maximum 100**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

Configuring Equal-Cost Multipath Load Sharing

Equal-cost multipath (ECMP) sets are formed when the router finds routing table entries for the same destination with equal cost. The router then balances traffic across these sets of equal-cost paths by using hashed mode.

Hashed Mode

Hashed mode uses hashing of source and destination addresses to determine which of the available paths in the ECMP set to use. Hashed mode is the default ECMP mode of operation.

Defining Maximum Paths

You can add routing table entries manually (as static routes), or they are formed as routers discover their neighbors and exchange routing tables (via OSPF, BGP, and other routing protocols).

The **maximum paths** command controls the maximum number of parallel routes that the routing protocol (BGP, IS-IS, OSPF, or RIP) can support.

maximum-paths

- Use to control the maximum number of parallel routes that the routing protocol supports.
- The maximum number of routes can be in the range 1–16 for BGP, IS-IS, OSPF, or RIP.
- Example

  host1(config-router)#**maximum-paths 2**

- Use the **no** version to restore the default value, 1 for BGP or 4 for IS-IS, OSPF, or RIP.

Fast Reroute Protection

If a link goes down, ECMP uses fast reroute protection to shift packet forwarding to use operational links, thereby decreasing packet loss. Fast reroute protection updates ECMP sets for the interface without having to wait for the route table update process. When the next route table update occurs, a new ECMP set can be added with fewer links or the route might point to a single next hop.

CAUTION: To provide ECMP fast reroute functionality in the event of an interface failure, the members of an equal cost multipath must be resolved to corresponding interfaces. If the member is an indirect next hop, the interface is obtained by using the forwarding equivalence class (FEC) to which the member points. This method of resolving members occurs only if the FEC, pointed to by the indirect next hop, is either an interface or a direct next hop.

An indirect next hop member is not resolved to an interface if it points to another indirect next hop or to an equal cost multipath. ECMP fast reroute functionality is not available if any interfaces that correspond to unresolved indirect next hop members go down.

If you modify an indirect next hop member to point to a different FEC (that is, a different interface, direct next hop, indirect next hop, or ECMP), the indirect next hop member is not resolved for the new changes.

Removing an IPv6 Configuration

To remove an IPv6 configuration from the virtual router, issue the **no ipv6** command.

no ipv6

- Use to remove IPv6 configuration from the virtual router.
- Example

  host1(config)#**no ipv6**

NOTE: The E-series router automatically starts IPv6 processing when you begin configuring an IPv6 interface. However, by issuing the **ipv6** command without using the **no** option, you can create an IPv6 processing instance with no IPv6 configuration.

Clearing IPv6 Routes

To clear dynamic IPv6 routes from the routing table, use the **clear ipv6 routes** command. To clear the routes for a specific IPv6 network, specify the IPv6 prefix. To clear all dynamic IPv6 routes, using the ***** (asterisk) option.

clear ipv6 routes

- Use to clear IPv6 routes.
- To clear routes in a specific IPv6 network, specify an IPv6 prefix.
- To clear all dynamic IPv6 routes, use the * (asterisk) option.
- Example

  host1(config)#**clear ipv6 routes ***

- There is no **no** version.

Creating Static IPv6 Neighbors

To create static IPv6 neighbors, use the **ipv6 neighbor** command.

ipv6 neighbor

- Use to create static IPv6 neighbors.
- Example

  host1(config)#**ipv6 neighbor 1::10 fastEthernet 1/0 0002.7dfa.0034**

- Use the **no** version of this command to delete the neighbor.

Clearing Dynamic IPv6 Neighbors

To clear dynamic IPv6 neighbors, use the **clear ipv6 neighbor** command. Using the **include-statics** keyword clears both dynamic neighbors and static neighbors. Using the **statics-only** keyword clears only IPv6 static neighbors.

clear ipv6 neighbors

- Use to clear all dynamic IPv6 neighbors.
- Use the **include-statics** keyword to clear both dynamic neighbors and static neighbors. Use the **statics-only** keyword to clear only IPv6 static neighbors.
- Example

  host1(config)#**clear ipv6 neighbors**

- There is no **no** version.

Monitoring IPv6

This section explains how to set an IPv6 statistics baseline and use the **show** commands to view your IPv6 configuration, monitor IPv6 interfaces and statistics, and view IPv6 neighbors. Many of these show commands also contain Neighbor Discovery information.

System Event Logs

To troubleshoot and monitor IPv6, use the following system event logs:

- ipv6General—IPv6 general information
- ipv6Interface—IPv6 interface events
- ipv6ProfileMgr—IPv6 profile manager events
- ipv6RouteTable—IPv6 routing table events
- ipv6Traffic—IPv6 frame transmit and receive events

For more information about using event logs, see the *JUNOSe System Event Logging Reference Guide*.

Establishing a Baseline

IPv6 statistics are stored in system counters. The only way to reset the system counters is to reboot the system. You can, however, establish a baseline for IPv6 statistics by setting a group of reference counters to zero (0).

baseline ipv6

- Use to set a baseline for IPv6 statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **udp** keyword to set a baseline for UDP statistics
- Use the **delta** keyword with IPv6 **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ipv6**

- There is no **no** version

baseline ipv6 interface

- Use to set a statistical baseline for a specified IPv6 interface.
- Example

  host1#**baseline ipv6 interface atm 2/0.100**

- There is no **no** version.

baseline tcp

- Use to set a statistics baseline for all (both IPv4 and IPv6) TCP statistics or for only IPv4 or IPv6 statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **ipv6** keyword to implement a baseline for only IPv6 statistics.
- Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.
- Example 1

  host1#**baseline tcp**

- Example 2

  host1#**baseline ipv6 tcp**

- There is no **no** version.

IPv6 show Commands

You can monitor the following aspects of IPv6 using **show ipv6** commands:

To Display	Command
General IPv6 information	**show ipv6**
IPv6 addresses	**show ipv6 address**
IPv6 forwarding table	**show ipv6 forwarding table slot**
IPv6 Interfaces	**show ipv6 interface**
IPv6 neighbors	**show ipv6 neighbors**
IPv6 profile information	**show ipv6 profile**
Active IPv6 protocol information	**show ipv6 protocols**
IPv6 route redistribution configuration	**show ipv6 redistribute**
IPv6 routes	**show ipv6 route**
IPv6 router advertisements received	**show ipv6 routers**
IPv6 static routes	**show ipv6 static**
IPv6 statistics/traffic	**show ipv6 traffic**
IPv6 UDP information	**show ipv6 udp statistics**
IPv6 license string	**show license ipv6**
IPv6 TCP information	**show tcp statistics** **show ipv6 tcp statistics**

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string that you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

show ipv6

- Use to display general IPv6 information.
- Example

```
host1#show ipv6
 Ipv6 Unicast Routing: Enabled
 Default hop limit: not specified
 Number of interfaces: 2
 Default interface source address/mask: fe80::90:1a00:210:fd0/128
```

show ipv6 address
show ipv6 interface

- Use to display detailed or summary information for a particular IPv6 address or interface or for all interfaces.
- The default for the **show ipv6 interface command** is all interface types and all interfaces.
- Use **brief** or **detail** keywords with the **show ipv6 interface command** to display different levels of information.
- Field descriptions
  - Description—Optional description for the interface or address specified
  - Network Protocols—Network protocols configured on this interface
  - Link local address—Local IPv6 address of this interface
  - Internet address—External address of this interface
  - IPv6 statistics Rcvd:
    - local destination—Frames with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IPv6 statistics Sent:
    - generated—Number of packets generated
    - no routes—Number of packets that could not be routed
    - discards—Number of packets that could not be routed that were discarded
  - ICMPv6 statistics Rcvd:
    - total—Total number of received packets
    - errors—Error packets received
    - destination unreach—Packets received with destination unreachable
    - admin unreach—Packets received because the destination was administratively unreachable (for example, the packet encountered a firewall filter)
    - parameter problem—Packets received with parameter errors
    - time exceeded—Packets received with time-to-live exceeded
    - pkt too big—Number of packet-too-big messages received that indicate a packet was too large to forward because of the allowed MTU size
    - redirects—Received packet redirects
    - echo requests—Echo request (ping) packets
    - echo replies—Echo replies received
    - rtr solicits—Number of received router solicitations

- rtr advertisements—Number of received router advertisements
- neighbor solicits—Number of received neighbor solicitations
- neighbor advertisements—Number or received neighbor advertisements
- Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests received from within a group to which the interface is assigned

- ICMPv6 statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received
  - destination unreach—Packets received with destination unreachable
  - admin unreach—Packets sent because the destination was administratively unreachable (for example, due to a firewall filter)
  - parameter problem—Packets received with parameter errors
  - time exceeded—Packets received with time-to-live exceeded
  - pkt too big—Number of packet-too-big messages sent because a received packet was too large to forward because of the allowed MTU size
  - redirects—Received packet redirects
  - echo requests—Echo request (ping) packets
  - echo replies—Echo replies received
  - rtr solicits—Number of sent router solicitations
  - rtr advertisements—Number of sent router advertisements
  - neighbor solicits—Number of sent neighbor solicitations
  - neighbor advertisements—Number of sent neighbor advertisements
  - Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests sent to a group of which the interface is assigned
- Operational MTU—Value of the MTU
- Administrative MTU—Value of the MTU if it has been administratively overridden using the configuration
- Operational speed—Speed of the interface
- Administrative speed—Value of the speed if it has been administratively overridden using the configuration
- Creation type—Method by which the interface was created (static or dynamic)
- ND reachable time—Amount of time (in milliseconds) that the neighbor is expected to remain reachable
- ND duplicate address detection attempts—Number of times that the router attempts to determine a duplicate address
- ND neighbor solicitation retransmission interval—Interval in which the router retransmits neighbor solicitations

- ND proxy—Indicates whether the router will reply to solicitations on behalf of a known neighbor
- ND RA source link layer—Indicates whether the RA includes the link layer
- ND RA interval—Interval (in seconds) of the neighbor discovery router advertisement
- ND RA lifetime—Lifetime (in seconds) of the neighbor discovery router advertisement
- ND RA managed flag—State of the neighbor discovery router advertisement managed flag
- ND RA other config flag—State of the neighbor discovery router advertisement other config flag
- ND RA advertising prefixes—Configured advertisement prefixes for neighbor discovery router advertisement
- In Received Packets, Bytes—Total number of packets and bytes received on this interface
  - Unicast Packets, Bytes—Unicast packets and bytes received on the IPv6 interface; link-local received multicast packets (non-multicast-routed frames) are counted as unicast packets
  - Multicast Packets, Bytes—Multicast packets and bytes received on the IPv6 interface which are then multicast-routed are counted as multicast packets
- In Total Dropped Packets, Bytes—Total number of inbound packets and bytes dropped on this interface
  - In Policed Packets—Packets that were received and dropped because of rate limits
  - In Invalid Source Address Packets—Packets received with invalid source address (for example, spoofed packets)
  - In Error Packets—Number of packets received with errors
  - In Discarded Packets—Packets received that were discarded for reasons other than rate limits, errors, and invalid source address
- Out Forwarded Packets, Bytes—Total number of packets and bytes that were sent from this interface
  - Unicast Packets, Bytes—Unicast packets and bytes that were sent from this interface
  - Multicast Routed Packets, Bytes—Multicast packets and bytes that were sent from this interface

- Out Total Dropped Packets—Total number of outbound packets and bytes dropped by this interface
  - Out Scheduler Dropped Packets, Bytes—Number of outbound packets and bytes dropped by the scheduler
  - Out Policed Packets, Bytes—Number of outbound packets and bytes dropped because of rate limits
  - Out Discarded Packets—Number of outbound packets that were discarded for reasons other than those dropped by the scheduler and those dropped because of rate limits
- IPv6 policy—Type (input, output, local-input) and name of policy
  - rate-limit-profile—Name of profile
  - classifier-group entry—Entry index
  - Committed—Number of packets and bytes conforming to the committed access rate
  - Conformed—Number of packets and bytes that exceed the committed access rate but conform to the peak access rate
  - Exceeded—Number of packets and bytes exceeding the peak access rate
- queue, traffic class, bound to ipv6—Queue and traffic class bound to the specified IPv6 interface
  - Queue length—Number of bytes in queue
  - Dropped committed packets, bytes—Total number of committed packets and bytes dropped by this interface
  - Dropped conformed packets, bytes—Total number of conformed packets and bytes dropped by this interface
  - Dropped exceeded packets, bytes—Total number of exceeded packets and bytes dropped by this interface

- Example 1

```
host1#show ipv6 address 5:1:1::2
FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce
  Internet address: 5:1:1::2/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 12, Bytes 1260
    Unicast Packets 5, Bytes 588
    Multicast Packets 7, Bytes 672
```

```
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 21, Bytes 2352
    Unicast Packets 21, Bytes 2352
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 2

```
host1#show ipv6 address detail 5:1:1::2
FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce

  Internet address: 5:1:1::2/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         3 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 5 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  ICMPv6 statistics:
    Rcvd:  12 total, 0 errors
           0 rtr solicits, 7 rtr advertisements
           1 neighbor solicits, 1 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  31 total, 0 errors
           0 rtr solicits, 16 rtr advertisements
           5 neighbor solicits, 5 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 12, Bytes 1260
    Unicast Packets 5, Bytes 588
    Multicast Packets 7, Bytes 672
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 3

```
host1#show ipv6 interface
null0 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44
  Unnumbered Interface: Corresponding Numbered Interface not specified or
removed
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
```

```
  Link local address: fe80::90:1a00:740:31ce
  Internet address: 5:1:1::2/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled

  In Received Packets 13, Bytes 1356
    Unicast Packets 5, Bytes 588
    Multicast Packets 8, Bytes 768
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd
  Internet address: 6:1:1::1/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
```

```
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

loopback5 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44
  Internet address: 10:1:1:0:290:1aff:fe40:1d44/64 (eui-64)
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

    IPv6 policy input ipv6InPol25
    rate-limit-profile Rlp2Mb classifier-group clgA entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp8Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy output ipv6PolOut2
    rate-limit-profile RlpOutA classifier-group clgB entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile RlpOutB
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy local-input ipv6PolLocIn5
    rate-limit-profile Rlp1Mb classifier-group clgC entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
```

```
    rate-limit-profile Rlp5Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 4

```
host1#show ipv6 interface FastEthernet 9/0.6
FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd
  Internet address: 6:1:1::1/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

    IPv6 policy input ipv6InPol25
    rate-limit-profile Rlp2Mb classifier-group clgA entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
```

```
    rate-limit-profile Rlp8Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy output ipv6PolOut2
    rate-limit-profile RlpOutA classifier-group clgB entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile RlpOutB
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy local-input ipv6PolLocIn5
    rate-limit-profile Rlp1Mb classifier-group clgC entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp5Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 5

```
host1#show ipv6 interface detail
null0 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44

  Unnumbered Interface: Corresponding Numbered Interface not specified or
removed
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  ICMPv6 statistics:
    Rcvd:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce

  Internet address: 5:1:1::2/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         3 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 5 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  ICMPv6 statistics:
    Rcvd:  13 total, 0 errors
           0 rtr solicits, 8 rtr advertisements
           1 neighbor solicits, 1 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  31 total, 0 errors
           0 rtr solicits, 16 rtr advertisements
           5 neighbor solicits, 5 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 13, Bytes 1356
    Unicast Packets 5, Bytes 588
    Multicast Packets 8, Bytes 768
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd

  Internet address: 6:1:1::1/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface
```

```
  ICMPv6 statistics:
    Rcvd:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
    Sent:  13 total, 0 errors
           0 rtr solicits, 9 rtr advertisements
           2 neighbor solicits, 2 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

loopback5 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44

  Internet address: 10:1:1:0:290:1aff:fe40:1d44/64 (eui-64)
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
```

```
Operational MTU 1500 Administrative MTU 0
Operational speed 100000000 Administrative speed 0
Creation type Static
Neighbor Discovery is disabled

ICMPv6 statistics:
  Rcvd:  0 total, 0 errors
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
         0 redirects
  Sent:  0 total, 0 errors
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
         0 redirects

In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Total Dropped Packets 0, Bytes 0
  In Policed Packets 0
  In Invalid Source Address Packets 0
  In Error Packets 0
  In Discarded Packets 0

Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Total Dropped Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0
  Out Discarded Packets 0
  IPv6 policy input ipv6InPol25
  rate-limit-profile Rlp2Mb classifier-group clgA entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile Rlp8Mb
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
IPv6 policy output ipv6PolOut2
  rate-limit-profile RlpOutA classifier-group clgB entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile RlpOutB
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
IPv6 policy local-input ipv6PolLocIn5
  rate-limit-profile Rlp1Mb classifier-group clgC entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile Rlp5Mb
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
```

```
    queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0
```

- Example 6

```
host1#show ipv6 interface brief
      Interface            IPv6-Address     Status  Protocol    Description
---------------------- -------------------- ------- -------- ------------------
null0                  Unnumbered           up      up
FastEthernet9/1.5      5:1:1::2/64          up      up       IPv6 interface in
                                                             Virtual Router Hop
                                                             5
FastEthernet9/0.6      6:1:1::1/64          up      up       IPv6 interface in
                                                             Virtual Router Hop
                                                             6
loopback5              10:1:1:0:290:1aff:fe up      up
                       40:1d44/64
```

show ipv6 forwarding-table slot

- Use to display details on the forwarding table for a specific line module only when IPv6 is configured on the router. These details include the memory used by each virtual router configured on the line module and free memory available on the module.
- The Load Errors field records any failed routing table distribution attempt as an error. Attempts can fail for many reasons during normal operation; a failed attempt does not necessarily indicate a problem. It is normal to see many Load Errors per day.
- If the Status field does not indicate Valid, then the routing table distribution has failed constantly for that VR. It is normal and appropriate behavior for the Status field to indicate Valid while the Load Error field increases daily.
- Field descriptions
  - Free Memory—Amount of routing table memory free on the line module, in kilobytes
  - Virtual Router—Name of the virtual routers configured on the line module
  - Memory (KB)—Amount of routing table memory consumed by the virtual router, in kilobytes
  - Load Errors—Count of errors made while loading the routing table on the line module
  - Status—Whether the routing table for the virtual router is valid
- Example

```
host1#show ipv6 forwarding-table slot 9
Free Memory = 32766 KB (99.99%)
         Virtual Router            Memory(KB)    Load Errors      Status
------------------------------ -------------- ------------- --------------
default                        -              -             Not Resident
1                              2              0             Valid
```

show ipv6 neighbors

- Use to display IPv6 Neighbor Discovery cache information static entries, dynamic entries, or both.
- Use the **static** keyword to display only static entries
- Use the **dynamic** keyword to display only dynamic entries
- Use the **summary** keyword to display summary information
- Field descriptions
  - Interface—Neighbor interface
  - IPv6-Address—IPv6 address for the interface
  - Type—Type of interface (dynamic, static)
  - Hardware Addr—Layer 2 address of the interface
  - State—State of the interface (delay, incomplete, probe, reachable, stale)
  - Age—Amount of time (in seconds) since the router contacted the neighbor
  - By type—List by neighbor type (global, link-local, anycast, and unknown)
  - By state—List by neighbor state (reachable, incomplete, stale, probe, delay, an init)
  - IPv6 address conflicts—Number of conflicts during or after duplicate address detection resolution
- Example 1

```
host1#show ipv6 neighbors
      Interface            IPv6-Address      Type   Hardware Addr  State  Age
---------------------- --------------------- ------- -------------- ----- -----
FastEthernet4/1        1::1                 dynamic 0090.1a40.05e5 reach 3
```

- Example 2

```
host1#show ipv6 neighbors summary
Total IPv6 neighbors: 7
By type: 5 global, 2 link-local, 0 anycast, 0 unknown
By state: 5 reachable, 0 incomplete, 2 stale, 0 probe, 0 delay, 0 init
IPv6 address conflicts: 0 during DAD resolution, 0 after DAD resolution
```

show ipv6 profile

- Use to display information about a specific IPv6 profile.
- Field descriptions
  - IPv6 profile—Profile name
  - Unnumbered interface—Specifier for the unnumbered interface or none if the interface is numbered
  - Router—Router name

- Access Route Addition—Enabled or disabled
- Source-Address Validation—Enabled or disabled
- Administrative MTU—MTU size

- Example

```
host1#show ipv6 profile foo
IPv6 profile : foo
Unnumbered interface on : loopback 0
Router                     : r1
Access Route Addition      : Enabled
Source-Address Validation  : Disabled
Administrative MTU         : 0
```

show ipv6 protocols

- Use to display configured protocols.
- Field descriptions
  - Local router ID—Router ID of the local router
  - Local AS—AS number of local router
  - Administrative state—Administrative state of the protocol
  - Operational state—Operational state of the protocol
  - Shutdown in overload state—Status of shutdown in an overload state
  - Default local preference—Default value for local preference
  - IGP synchronization—Indicates whether synchronization is enabled or disabled
  - Default originate—Indicates whether network 0.0.0.0 is redistributed into BGP
  - Auto summary—Status of autosummary
  - Always compare MED—Status of always compare MED
  - Compare MED within confederation—Status of compare MED within a confederation
  - Advertise inactive routes—Status of Advertise inactive routes
  - Advertise best external router to internal peers—Status of Advertise best external router to internal peers
  - Enforce first AS—Status of Enforce first AS
  - Missing MED as worst—Status of Missing MED as worst
  - Route flap dampening—Status of route dampening
  - Log neighbor changes—Status of Log neighbor changes
  - Fast External Fallover—Status of Fast External Fallover
  - Maximum received AS-path length—Maximum AS-path length received
  - BGP administrative distances—External, internal, and local BGP administrative distances
  - Client-to-client reflection—Whether client-to-client reflection is configured
  - Cluster ID—Cluster IDs

- Route-target filter—Status of Route-target filter
- Default IPv4-unicast—Status of Default IPv4-unicast
- Local-RIB version—RIB version
- Local-FIB version—FIB version
- Neighbor(s)—BGP neighbors (if configured)
- Networks for which routing is occurring
- Aggregate Generation for Unicast Routes

- Example 1

```
host1#show ipv6 protocols
Routing Protocol is "bgp 100"
  Local router ID 1.1.1.1, local AS 100
  Administrative state is Start
  Operational state is Up
  Shutdown in overload state is disabled
  Default local preference is 100
  IGP synchronization is enabled
  Default originate is disabled
  Auto summary is enabled
  Always compare MED is disabled
  Compare MED within confederation is disabled
  Advertise inactive routes is disabled
  Advertise best external route to internal peers is disabled
  Enforce first AS is disabled
  Missing MED as worst is disabled
  Route flap dampening is disabled
  Log neighbor changes is disabled
  Fast External Fallover is disabled
  No maximum received AS-path length
  BGP administrative distances are 20 (ext), 200 (int), and 200 (local)
  Client-to-client reflection is enabled
  Cluster ID is 1.1.1.1
  Route-target filter is enabled
  Default IPv4-unicast is enabled
  Local-RIB version 8. FIB version 8.
  Neighbor(s):
    No neighbors are configured
  Routing for Networks:
  Aggregate Generation for Unicast Routes:
```

- Example 2

```
host1#show ipv6 protocols summary
bgp 100
```

show ipv6 redistribute

- Use to display configured route redistribution policy.
- Field descriptions
  - To—Protocol that routes are distributed into
  - From—Protocol that routes are distributed from
  - status—Redistribution status
  - route map name—Name of the route map
- Example

```
host1#show ipv6 redistribute
To bgp, From static is enabled with route map foo
To bgp, From connected is enabled without a route map
```

show ipv6 route

- Use to display the current state of the routing table, including routes not used for forwarding.
- You can display all routes, a specific route, detailed information about all or a specific route, or summary counters for the routing table.
- Field descriptions
  - Prefix—IPv6 address prefix
  - Length—Prefix length
- Type—Protocol type (possible route types include: Bgp, Connect, Idrp, Igrp, Invalid, Isis, Ndisc, Ospf, Other, Rip, Static)
  - Dst (or Distance)—Administrative distance for the route
  - Met (or Metric)—Number of hops
  - Intf (or Interface)—Interface type and interface specifier
  - NextHop—The configured next hop address for this interface
  - IfIndex—An autogenerated value for the next hop interface
- Example 1

```
host1#show ipv6 route
         Prefix/Length             Type   Dst/Met         Intf
------------------------------- ------- ------- --------------------
1::/16                          Connect 0/0     loopback1
5::/64                          Connect 0/0     ATM4/0.15
6::/64                          Static  1/0     ATM4/0.15
2003::/16                       Static  1/0     ATM4/0.15
```

- Example 2

```
host1#show ipv6 route summary
Unicast routes:
8 total routes, 576 bytes in route entries
0 isis routes
0 rip routes
3 static routes
2 connected routes
1 bgp routes
0 ospf routes
2 other internal routes
```

```
0 access routes
0 internally created access host routes

Last route added/deleted: 2::4/128 by BGP
At MON FEB 04 2008 14:18:25 UTC

Unicast routes used only for Multicast RPF check:
0 total routes, 0 bytes in route entries
0 isis routes
0 rip routes
0 static routes
0 connected routes
0 bgp routes
0 ospf routes
0 other internal routes
0 access routes
0 internally created access host routes
0 mbgp routes
0 dvmrp routes

Last route added/deleted: null by Invalid
At MON FEB 04 2008 14:18:04 UTC

MPLS tunnel routes (not used for forwarding):
3 total routes, 216 bytes in route entries
1 bgp tunnel routes
1 ldp tunnel routes
1 rsvp tunnel routes

Last route added/deleted: 2::4/128 by BGP Tunnel
At MON FEB 04 2008 14:18:26 UTC
```

- Example 3

```
host1#show ipv6 route 5::/64 detail
5::/64 Type:local  Distance:0  Metric:0
        NextHop: 1::2  IntfIndex 10007 Intf ATM4/0.15
```

show ipv6 routers

- Use to display IPv6 router advertisement information received.
- Use the conflicts keyword to display router advertisements that differ from the advertisements configured
- Field descriptions
  - Route—Router for which this information applies
  - Hops—Number of hops that the router uses in router advertisements
  - Lifetime—Lifetime (in seconds) of the neighbor discovery router advertisement
  - AddrFlag—State of the neighbor discovery router advertisement managed flag
  - OtherFlag—State of the neighbor discovery router advertisement other config flag
  - Reachable time—Amount of time (in milliseconds) that the neighbor is expected to remain reachable

  - Retransmit time—Interval in which the router retransmits neighbor solicitations
  - Prefix—IPv6 network number to include in router advertisements
  - Autoconfig—When present, indicates that local host links use the specified prefix for IPv6 autoconfiguration
  - Valid lifetime—Amount of time in seconds that the router advertises the IPv6 prefix as valid
  - preferred lifetime—Amount of time in seconds that the router advertises the specified IPv6 prefix as preferred
- Example 1

```
host1#show ipv6 routers
Router FE80::83B3:60A4 on FastEthernet2/0, last update 3 min
  Hops 0, Lifetime 6000 sec, AddrFlag=0, OtherFlag=0
  Reachable time 0 msec, Retransmit time 0 msec
  Prefix 3FFE:C00:8007::800:207C:4E37/96 autoconfig
    Valid lifetime -1, preferred lifetime -1

Router FE80::290:27FF:FE8C:B709 on FastEthernet2/1, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0
  Reachable time 0 msec, Retransmit time 0 msec
```

- Example 2

```
host1#show ipv6 routers conflicts
Router FE80::203:FDFF:FE34:7039 on FastEthernet1/0, last update 1 min,
CONFLICT
   Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0
   Reachable time 0 msec, Retransmit time 0 msec
   Prefix 2003::/64 onlink autoconfig
     Valid lifetime -1, preferred lifetime -1
```

show ipv6 static

- Use to display the status of static routes in the routing table.
- You can specify an IP mask that filters specific routes.
- Field descriptions
  - Prefix—IP address prefix
  - Length—Prefix length
  - Next Hop—IP address of the next hop
  - Dst—Administrative distance of the route
  - Met—Number of hops
  - Interface—Interface type and interface specifier
- Example

```
host1#show ipv6 static
     Prefix/Length               NextHop          Dst/Met      Interface
------------------------ ------------------------ ------- --------------------
6::/64                   5::2                     1/0     ATM4/0.15
2003::/16                5::1                     1/0     ATM4/0.15
```

show ipv6 traffic

- Use to display statistics about IPv6 traffic.
- Field descriptions
  - IPv6 statistics Rcvd:
    - total—Total number of packets received
    - local destination—Number of packets received with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IPv6 statistics Sent:
    - forwarded—Number of packets forwarded
    - generated—Number of packets generated
    - out disc—Number of packets that could not be routed that were discarded
  - IPv6 statistics Mcast:
    - received—Number of multicast packets received
    - forwarded—Number of multicast packets forwarded
  - IPv6 statistics (Routes)—Number of routes currently in the routing table
  - ICMPv6 statistics Rcvd:
    - total—Total number of received packets
    - errors—Error packets received
    - destination unreach—Packets received with destination unreachable
    - admin unreach—Packets received because the destination was administratively unreachable (for example, the packet encountered a firewall filter)
    - parameter problem—Packets received with parameter errors
    - time exceeded—Packets received with time-to-live exceeded
    - pkt too big—Number of packet-too-big messages received that indicate a packet was too large to forward because of the allowed MTU size
    - redirects—Received packet redirects
    - echo requests—Echo request (ping) packets
    - echo replies—Echo replies received
    - rtr solicits—Number of received router solicitations
    - rtr advertisements—Number of received router advertisements

- neighbor solicits—Number of received neighbor solicitations
- neighbor advertisements—Number of received neighbor advertisements
- Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests received from within a group to which the interface is assigned

- ICMP statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received
  - destination unreach—Packets received with destination unreachable
  - admin unreach—Packets sent because the destination was administratively unreachable (for example, due to a firewall filter)
  - parameter problem—Packets received with parameter errors
  - time exceeded—Packets received with time-to-live exceeded
  - pkt too big—Number of packet-too-big messages sent because a received packet was too large to forward because of the allowed MTU size
  - redirects—Received packet redirects
  - echo requests—Echo request (ping) packets
  - echo replies—Echo replies received
  - rtr solicits—Number of sent router solicitations
  - rtr advertisements—Number of sent router advertisements
  - neighbor solicits—Number of sent neighbor solicitations
  - neighbor advertisements—Number of sent neighbor advertisements
  - Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests sent to a group to which the interface is assigned
- UDP Statistics Rcvd:
  - total—Total number of received packets
  - checksum errors—Checksum error packets received
  - no port—No port error packets received
- UDP Statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received

- Example

```
host1#show ipv6 traffic
IPv6 statistics:
  Rcvd:  0 total, 0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 forwarded, 0 generated
         0 out disc
  Mcast: 0 received 0 forwarded
  Routes: 7 in routing table

ICMPv6 statistics:
  Rcvd:  0 total, 0 errors
         0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 redirects
         0 echo requests, 0 echo replies
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
  Sent:  3 total, 0 errors
         0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 redirects
         0 echo requests, 0 echo replies
         0 rtr solicits, 0 rtr advertisements
         2 neighbor solicits, 1 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions

UDP Statistics:
  Rcvd: 0 total, 0 checksum errors, 0 no port
  Sent: 0 total, 0 errors
```

show ipv6 udp statistics

- Use to display IPv6 UDP statistics.
- Example

```
host1#show ipv6 udp statistics
UDP Statistics:
  Rcvd: 0 total, 0 checksum errors, 0 no port
  Sent: 0 total, 0 errors
```

show license ipv6

- Use to display the IPv6 license key configured on the router.
- Example

```
host1#show license ipv6
Ipv6 license is ipv6_license
```

show tcp statistics

- Use to display all TCP statistics (both IPv4 and IPv6).
- Baselining is supported for this command.
- Use the **ip** keyword to display only IPv4 statistics.
- Use the **ipv6** keyword to display only IPv6 statistics.

- Use the **brief** keyword to display summary information or the **detailed** keyword to display extensive information.
- Use the **diagnostic** keyword to display diagnostic information collected on the TCP statistics in addition to the detailed information. This command shows information only for the connections that are active within the context of the VR in which you issue the command.
- Field descriptions
  - TCP Global Statistics Connections:
    - attempted—Number of outgoing TCP connections attempted
    - accepted—Number of incoming TCP connections accepted
    - established—Number of TCP connections established
  - TCP Global Statistics Rcvd:
    - total pkts—Total number of packets received
    - in-sequence pkts—Number of packets received in sequence
    - bytes—Number of bytes received
    - chksum err pkts—Number of checksum error packets received
    - authentication err pkts—Number of authentication error packets received
    - bad offset pkts—Number of bad offset packets received
    - short pkts—Number of short packets received
    - duplicate pkts—Number of duplicate packets received
    - out of order pkts—Number of packets received out of order
  - TCP Global Statistics Sent:
    - total pkts—Total number of packets sent
    - data pkts—Number of data packets sent
    - bytes—Number of bytes sent
    - retransmitted pkts—Number of packets retransmitted
    - retransmitted bytes—Number of bytes retransmitted

- Global Diagnostic Data Unknown Connection log—Includes the following global statistics:
  - Source address/port – local port—Shows the 32 most recent TCP connection attempts that were rejected, including the remote node's IP or IPv6 address and port, the local port for the connection attempt, and the number of identical attempts that have been received on that port in a row. The reason for rejection is not given. This information may be useful in tracking down DoS attacks.
  - # connection-reqs rejected—Total number of connection attempts that have been rejected
  - # connection-reqs pending—Current number of connection attempts that are pending, awaiting additional data from the peer
  - # sonewconn calls that fail—Number of calls to sonewconn that have failed. This statistic often indicates that either a socket connection limit has been reached or that there was no memory to hold the socket data structures.
- TCP Session Statistics
  - Local addr—Local address of the TCP connection
  - Local port—Local port number of the TCP connection
  - Remote addr—Remote address of the TCP connection
  - Remote port—Remote port number of the TCP connection
  - State—Current state of the TCP connection
  - Authentication—Authentication status of the TCP connection
- TCP Session Statistics Sent:
  - total pkts—Total number of packets sent on the TCP connection
  - data pkts—Number of data packets sent on the TCP connection
  - bytes—Number of bytes sent on the TCP connection
  - retransmitted pkts—Number of packets retransmitted on the TCP connection
  - retransmitted bytes—Number of bytes retransmitted on the TCP connection
- TCP Session Statistics Rcvd:
  - total pkts—Total number of packets received on the TCP connection
  - in-sequence pkts—Number of packets received in sequence on the TCP connection
  - bytes—Number of bytes received on the TCP connection
  - chksum err pkts—Number of checksum error packets received on the TCP connection
  - bad offset pkts—Number of bad offset packets received on the TCP connection

  - short pkts—Number of short packets received on the TCP connection
  - duplicate pkts—Number of duplicate packets received on the TCP connection
  - out of order pkts—Number of packets received out of order on the TCP connection
- Diagnostics: PRU_ Operations counters—Number of calls for each of the indicated PRU_operations within the TCP service API. These are per-connection statistics.
- Wildcard Matches—Number of packets received that matched this TCP connection due to wildcard matching. Matching is expected for listening server connections, such as Telnet, but is not expected for established connections. This is a per-connection statistic.
- Rcv'd Packets after connection closed—Number of packets received on the connection after the connection has been closed (and before the data structure gets removed). This is a per-connection statistic.
- Connect request rejected—Number of times an incoming connection request was not approved. This is a per-connection statistic.
- Connect request approval pending—Number of times that an incoming connection request was held pending, waiting for a subsequent packet. This is a per-connection statistic.
- New soconnect failed—Number of times a SONEWCONN() was tried on a listening connection and failed. This is a per-connection statistic.
- # Write-Wakeups—Number of times a “write wakeup” occurred on the connection. This is a per-connection statistic.
- # Read wakeups—Number of times a “read wakeup” occurred on the connection. This is a per-connection statistic.
- # receives after close—Number of packets received with data after the connection entered the close-wait state. This is a per-connection statistic.
- Retransmit timer—Current value of the retransmit timer
- Persistence timer—Current value of the persistence timer
- Keepalive timer—Current value of the keepalive timer
- 2MSL timer—Current value of the 2MSL (max segment lifetime) timer
- tcpDisconnect()s—Number of times BsdTcp::tcpDisconnect() was called. This is a per-connection statistic.
- keep T/O pre-estab—Number of times the keepalive timer expired before the connection reached the established state. This is a per-connection statistic.
- tcpkeeptimeo_idle—Number of times the keepalive timer popped, but no keepalive was sent because of connection idle-time considerations. This is a per-connection statistic.

- TCP Connection Event Log (most recent at bottom)—Event log for the TCP connection. It shows the last 32 events that occurred on the connection. The most recent event is at the bottom of the list. This is per-connection data.
  - TCPS_ELOG_PRU_ATTACH
  - TCPS_ELOG_PRU_BIND

The following events can be recorded:

Fast Timeout	Did a PRU_CONNECT
2MSL Timeout	Did a PRU_CONNECT2
Retransmit Timeout	Did a PRU_DISCONNECT
Persist Timeout	Did a PRU_ACCEPT
Received FIN packet	Did a PRU_SHUTDOWN
Received SYN packet	Did a PRU_RCVD
Received Retransmission	Did a PRU_SEND
Transmit a FIN packet	Did a PRU_ABORT
Transmit a SYN packet	Did a PRU_SENSE
Retransmit a packet	Did a PRU_RCVOOB
Did a PRU_ATTACH	Did a PRU_SENDOOB
Did a PRU_DETACH	Did a PRU_SOCKADDR
Did a PRU_BIND	Did a PRU_PEERADDR
Did a PRU_LISTEN	The keepalive timer popped. An 8-bit argument that describes how the timer was handled: ■ Ignored because the session was not established (that is, not in the OPEN state) ■ Ignored due to idle-timeout considerations ■ A packet was sent ■ Ignored because the connection did not have the keepalive option set OR the connection was in the process of closing

- RST/SYN-Ack DoS Protection—Specifies when this function is enabled
  - RSTs acked—Number of RSTs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus RSTs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - SYNs acked—Number of SYNs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus SYNs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - Data Insertions rejected—Number of packets received and dropped because they are believed to have been inserted by an attacker

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been rejected if the protections were enabled. Providing this information can help determine whether attacks are occurring.

- PMTUD information—Information regarding path MTU discovery
  - PMTUD—Status of path MTU discovery on the virtual router: enabled or disabled
  - Administrative Minimum MTU—Minimum MTU that is enabled on any connection; a value of “none” indicates that the minimum is zero (0)
  - Administrative Maximum MTU—Maximum MTU that is enabled on any connection; a value of “none” indicates that the maximum is 65535
  - Timer 1—Amount of time the virtual router waits after receiving an ICMP Too Big message before attempting to increase the path MTU
  - Timer 2—Amount of time the virtual router waits after successfully increasing the MTU before attempting to increase it more
  - # ICMP TooBigs—Number of ICMP Too Big messages that the router has received. When PMTU is disabled, this counter does not increase.
  - # ICMP TooBigs for unk. connection—Number of ICMP Too Big messages that the router has received for TCP connections that do not exist. When PMTU is disabled, this counter does not increase.

- PMTU Increase Attempts—Number of attempts the router has made to increase the PMTU
- Black Hole Detect Threshold—Number of successive transmissions that must occur on a connection before that connection treats retransmissions as indications that something is wrong
- Override MSS—MSS that is advertised to peers, overriding the MSS that is derived from the interface MTU. This line does not appear in the output if you do not set the value.

- MTU/MSS information—Information regarding path MTU/MSS
  - PMTU—Status of MTU/MSS on this virtual router: enabled or disabled
  - MSS in effect—MSS currently being used for transmission to the peer. This number changes while various network events occur to cause the router to increase or decrease its estimate of the MSS.
  - Calculated MSS to peer—MSS that path MTU discovery has calculated (if PMTUD is enabled) to the peer
  - MSS received from peer—MSS that the peer received in a TCP MSS option. If no option is received, the value is zero (0).
  - Application set MSS—MSS that an application might have set for the connection
  - Xmit Interface MSS—MSS for the interface used to transmit packets to the peer; calculated as the interface MTU minus the size of the TCP and IP headers.
  - MSS Sent to Peer—MSS that has been advertised to the peer
  - "ICMP DestUn, Frag Req'd and DF Set" messages—Number of ICMP "Destination Unreachable: Fragmentation Required and DF set" messages that the router has received
  - Number of attempts to increase PMTU—Number of times the router has attempted to increase the PMTU by probing with a packet that is larger than the known MTU
  - Time to next increase attempt—Amount of time, in seconds, until the router retries to increase the MTU
  - Black Hole Detection State—State of the black hole detection mechanism: none, detecting, probable, or unknown
- Out-of-Order Packet Queue Information—Information regarding packet queue buffers
  - Buffers Outstanding—Number of buffers currently on the connection reordering queue
  - High Water—Most buffers that have ever been on the connection reordering queue
  - Buffers discarded—Number of buffers that were discarded because keeping them would have exceeded the connection maximum
- TCP PAWS is [enabled/disabled]—Status of the TCP PAWS option; enabled indicates that PAWS is functioning normally (default mode) for TCP segments; disabled indicates that PAWS is disabled for TCP segments

- Example 1

```
host1#show ipv6 tcp statistics
TCP Global Statistics:
  Connections: 7358 attempted, 4 accepted, 7362 established
        0 dropped, 14718 closed
  Rcvd: 75923 total pkts, 53608 in-sequence pkts, 3120303 bytes
        0 chksum err pkts, 0 authentication err pkts, 0 bad offset pkts
        0 short pkts, 0 duplicate pkts, 0 out of order pkts
  Sent: 82352 total pkts, 44404 data pkts, 657095 bytes
        34 retransmitted pkts, 487 retransmitted bytes

TCP Session Statistics:
  Local addr: 0.0.0.0, Local port: 23
  Remote addr: 0.0.0.0, Remote port: 0
  State: LISTEN Authentication: None
  Rcvd: 4 total pkts, 0 in-sequence pkts, 0 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 0 total pkts, 0 data pkts, 0 bytes
        0 retransmitted pkts, 0 retransmitted bytes

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data pkts, 2304 bytes
        0 retransmitted pkts, 0 retransmitted bytes
 Local addr: 192.168.1.250, Local port: 23
  Remote addr: 192.168.1.139, Remote port: 1038
  State: ESTABLISHED Authentication: None
  Rcvd: 295 total pkts, 159 in-sequence pkts, 299 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 281 total pkts, 210 data pkts, 3089 bytes
        0 retransmitted pkts, 0 retransmitted bytes
```

- Example 2—Additional fields displayed by **diagnostic** keyword

```
host1#show tcp statistics diagnostic
...
Global Diagnostic Data
    Unknown Connection log
Source address/port -> local port
         128.127.126.125/124 -> 8080  count: 3
         111.111.111.111/222 -> 3333  count: 4
# connection-reqs rejected: 0
    # connection-reqs pending: 0
    # sonewconn calls that fail: 0
...
```

```
  Diagnostics:
    PRU_ Operations counters:
      PRU_ATTACH: 0
      PRU_DETACH: 0
      PRU_BIND: 1
      PRU_LISTEN: 1
      PRU_CONNECT: 0
      PRU_ACCEPT: 0
      PRU_DISCONNECT: 0
      PRU_SHUTDOWN: 0
      PRU_RCVD: 0
      PRU_SEND: 0
      PRU_ABORT: 0
      PRU_CONTROL: 0
      PRU_SENSE: 0
      PRU_RCVOOB: 0
      PRU_SENDOOB: 0
      PRU_SOCKADDR: 0
      PRU_PEERADDR: 0
      PRU_CONNECT2: 0
      PRU_FASTTIMO: 0
      PRU_SLOWTIMO: 0
      PRU_PROTORCV: 0
      PRU_PROTOSEND: 0
    Wildcard Matches: 2
    Rcv'd Packets after connection closed: 0
    Connect request rejected: 0
    Connect request approval pending 0
    New soconnect failed 0
    # Write-Wakeups: 0
    # Read wakeups 0
    # receives after close 0
    Retransmit timer: 0
    Persistance timer: 0
    Keepalive timer: 0
    2MSL timer: 0
    tcpDisconnect()s: 0
    keep T/O pre-estab: 0
    tcpkeeptimeo_idle: 0
...
TCP Connection Event Log (most recent at bottom)
      TCPS_ELOG_PRU_ATTACH
      TCPS_ELOG_PRU_BIND
```

- Example 3—Additional fields displayed by **detailed** keyword

```
host1#show tcp statistics detailed
...

RST/SYN-Ack Protection is: ENABLED
    RSTs acked: 0
    ...Bogus RSTs: 0
    SYNs acked: 0
    ...Bogus SYNs: 0
    Data Insertions rejected: 0
PMTUD Information:      PMTUD: ENABLED
    Administrative Minimum MTU: 512
    Administrative Maximum MTU: none
    Timer 1: 10 minutes
    Timer 2: 2 minutes
```

```
    # ICMP TooBigs: 0
    # ICMP TooBigs for unk. connection: 0
    PMTU Increase Attempts: 17
    Black Hole Detect Threshold: 50 retransmissions
...
MTU/MSS Information
    ENABLED on this connection
    MSS in effect: 536
    Calculated MSS to peer: 536
    MSS received from peer: 0
    Application set MSS: 0
    Xmit Interface MSS: 0
    MSS Sent to Peer: 0
    "ICMP DestUn, Frag Req'd and DF Set" messages: 0
    Number of attempts to increase PMTU: 0
    Time to next increase attempt: 0 seconds
    Black Hole Detection State: none
...
Out-of-order Packet Queue Information

  Buffers Outstanding: 25
           High Water: 28
    Buffers discarded: 15
...
TCP-Paws is disabled
```