

# Network Stack IPv4 and IPv6 Commands

This chapter describes the commands available on the NCS 5000 routers Cisco IOS XR software to configure and monitor features related to IP Version 4 (IPv4) and IP Version 6 (IPv6).

For detailed information about network stack concepts, configuration tasks, and examples, refer to the *IP Addresses and Services Configuration Guide for Cisco NCS 5000 Series Routers*.


- clear ipv6 neighbors, on page 3
- icmp ipv4 rate-limit unreachable, on page 4
- ipv4 address (network), on page 5
- ipv4 assembler max-packets, on page 7
- ipv4 assembler timeout, on page 8
- ipv4 mask-reply, on page 9
- ipv4 conflict-policy, on page 10
- ipv4 mtu, on page 11
- ipv4 unnumbered (point-to-point), on page 13
- ipv6 address, on page 15
- ipv6 assembler, on page 17
- ipv6 conflict-policy, on page 18
- ipv6 address link-local, on page 19
- ipv6 enable , on page 21
- ipv6 hop-limit, on page 23
- ipv6 icmp error-interval, on page 24
- ipv6 mtu, on page 26
- ipv6 nd dad attempts , on page 28
- ipv6 nd managed-config-flag, on page 31
- ipv6 nd ns-interval, on page 33
- ipv6 nd other-config-flag , on page 35
- ipv6 nd prefix, on page 37
- ipv6 nd ra-interval, on page 39
- ipv6 nd ra-lifetime, on page 41
- ipv6 nd reachable-time, on page 42
- ipv6 nd redirects, on page 44
- ipv6 nd scavenge-timeout, on page 45
- ipv6 nd suppress-ra, on page 46
- ipv6 neighbor, on page 47

- ipv6 path-mtu enable, on page 49
- ipv6 path-mtu timeout, on page 50
- ipv6 source-route, on page 51
- ipv6 tcp-mss-adjust, on page 52
- ipv6 virtual address, on page 53
- show arm conflicts, on page 55
- show arm registrations producers, on page 57
- show arm router-ids, on page 59
- show arm summary, on page 60
- show ipv4 interface, on page 61
- show ipv4 traffic, on page 64
- show ipv6 interface , on page 66
- show ipv6 neighbors, on page 70
- show ipv6 neighbors summary , on page 73
- show ipv6 path-mtu, on page 74
- show ipv6 traffic, on page 75

# clear ipv6 neighbors

To delete all entries in the IPv6 neighbor discovery cache, except static entries, use the **clear ipv6 neighbors** command in XR EXEC mode.

**clear ipv6 neighbors** [**location** *node-id*]

## Syntax Description

| | |
|---|---|
| **location** *node-id* | (Optional) The designated node. The *node-id* argument is entered in the *rack/slot/module* notation. |

## Command Default

None

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

If the location option is specified, only the neighbor entries specified in the **location** *node-id* keyword and argument are cleared.

## Task ID

| Task ID | Operations |
|---|---|
| network | read, write |
| IPv6 | execute |

## Examples

In the following example, only the highlighted entry is deleted:

```
RP/0/RP0/CPU0:router# clear ipv6 neighbors ?
location specify a node name

RP/0/RP0/CPU0:router# show ipv6 neighbor

IPv6 Address Age Link-layer Addr State Interface
8888::3 - 1234.2345.9877 REACH tenGigE0/0/0/0
8888::8 - 1234.2345.9877 REACH tenGigE0/0/0/0
fe80::205:1ff:fe9f:6400 1335 0005.019f.6400 STALE tenGigE0/0/0/0
fe80::206:d6ff:fece:3808 1482 0006.d6ce.3808 STALE tenGigE0/0/0/0
fe80::200:11ff:fe11:1112 1533 0000.1111.1112 STALE tenGigE0/0/0/2

RP/0/RP0/CPU0:router# clear ipv6 neighbors location 0/2/0
RP/0/RP0/CPU0:router# show ipv6 neighbor

IPv6 Address Age Link-layer Addr State Interface
8888::3 - 1234.2345.9877 REACH tenGigE0/0/0/0
8888::8 - 1234.2345.9877 REACH tenGigE0/0/0/0
fe80::205:1ff:fe9f:6400 1387 0005.019f.6400 STALE tenGigE0/0/0/0
fe80::206:d6ff:fece:3808 1534 0006.d6ce.3808 STALE tenGigE0/0/0/0
```

# icmp ipv4 rate-limit unreachable

To limit the rate that IPv4 Internet Control Message Protocol (ICMP) destination unreachable messages are generated, use the **icmp ipv4 rate-limit unreachable** command in XR Config mode. To remove the rate limit, use the **no** form of this command.

**icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*
**no icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*

## Syntax Description

| | |
|---|---|
| **DF** | (Optional) Limits the rate at which ICMP destination unreachable messages are sent when code 4 fragmentation is needed and data fragmentation is (DF) set, as specified in the IP header of the ICMP destination unreachable message. |
| *milliseconds* | Time period (in milliseconds) between the sending of ICMP destination unreachable messages. Range is 1 to 4294967295. |

## Command Default

The default value is one ICMP destination unreachable message every 500 milliseconds.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The Cisco IOS XR software maintains two timers: one for general destination unreachable messages and one for DF destination unreachable messages. Both share the same time limits and defaults. If the **DF** option is not configured, the **icmp ipv4 rate-limit unreachable** command sets the time values for DF destination unreachable messages. If the **DF** option is configured, its time values remain independent from those of general destination unreachable messages.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |

## Examples

The following example shows how to set the time interval for the ICMP destination unreachable message to be generated at a minimum interval of 10 ms:

```
RP/0/RP0/CPU0:router(config)# icmp ipv4 rate-limit unreachable 10
```

# ipv4 address (network)

To set a primary or secondary IPv4 address for an interface, use the **ipv4 address** command in interface configuration mode. To remove an IPv4 address, use the **no** form of this command.

**ipv4 address** *ipv4-address mask* [**secondary**] [**route-tag** *route-tag value*]
**no ipv4 address** *ipv4-address mask* [**secondary**] [**route-tag** *route-tag value*]

## Syntax Description

| | |
|---|---|
| **ipv4-address** | IPv4 address. |
| *mask* | Mask for the associated IP subnet. The network mask can be specified in either of two ways:<br>• The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address.<br>• The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address. |
| **secondary** | (Optional) Specifies that the configured address is a secondary IPv4 address. If this keyword is omitted, the configured address is the primary IPv4 address. |
| **route-tag** | (Optional) Specifies that the configured address has a route tag to be associated with it. |
| *route-tag value* | (Optional) Value of the route tag. Range is 1 to 4294967295. |

## Command Default

No IPv4 address is defined for the interface.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

An interface can have one primary IPv4 address and multiple secondary IPv4 addresses. Packets generated by the software always use the primary IPv4 address. Therefore, all networking devices on a segment should share the same primary network number.



**Note** The same IPv4 address configured on two different interfaces causes an error message to display that indicates the conflict. The interface located in the highest rack, slot, module, instance, and port is disabled.

Hosts can determine subnet masks using the IPv4 Internet Control Message Protocol (ICMP) mask request message. Networking devices respond to this request with an ICMP mask reply message.

You can disable IPv4 processing on a particular interface by removing its IPv4 address with the **no ipv4 address** command. If the software detects another host using one of its IPv4 addresses, it will display an error message on the console.

The optional **secondary** keyword allows you to specify an unlimited number of secondary addresses. Secondary addresses are treated like primary addresses, except that the system never generates datagrams other than routing updates with secondary source addresses. IPv4 broadcasts and Address Resolution Protocol (ARP) requests are handled properly, as are interface routes in the IP routing table.

Secondary IPv4 addresses can be used in a variety of situations. The following are the most common applications:

- There may not be enough host addresses for a particular network segment. For example, your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you need to have 300 host addresses. Using secondary IPv4 addresses on the networking devices allows you to have two logical subnets using one physical subnet.
- Many older networks were built using Level 2 bridges. The judicious use of secondary addresses can aid in the transition to a subnetted, router-based network. Routers on an older, bridged segment can be easily made aware that there are many subnets on that segment.

The route-tag feature attaches a tag to all IPv4 addresses. The tag is propagated from the Management Agents (MA) to the Address Repository Managers (RPM) to routing protocols, thus enabling the user to control the redistribution of connected routes by looking at the route tags via RPL scripts.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |

## Examples

The following example shows how to set 192.168.1.27 as the primary address and 192.168.7.17 and 192.168.8.17 as the secondary addresses on tenGigE interface 0/0/0/1:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.7.17 255.255.255.0 secondary
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.8.17 255.255.255.0 secondary
```

# ipv4 assembler max-packets

To configure the maximum number of packets that are allowed in assembly queues, use the **ipv4 assembler max-packets** command in XR Config mode. To disable this feature, use the **no** from of this command.

**ipv4 assembler max-packets** *percentage value*
**no ipv4 assembler max-packets** *percentage value*

## Syntax Description

| | |
|---|---|
| *percentage value* | Percentage of total packets available in the system. The range is from 1 to 50. |

## Command Default

None

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |

## Examples

The following example shows how to configure the maximum number of packets for the assembly queue:

```
RP/0/RP0/CPU0:router(config)# ipv4 assembler max-packets 35
```

# ipv4 assembler timeout

To configure the number of seconds an assembly queue can hold before a timeout occurs, use the **ipv4 assembler timeout** command in XR Config mode. To disable this feature, use the **no** form of this command.

**ipv4 assembler timeout** *seconds*
**no ipv4 assembler timeout** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | Number of seconds an assembly queue can hold before a timeout occurs. The range is from 1 to 120. |

## Command Default

None

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |

## Examples

The following example shows how to configure an assembly queue before a timeout occurs:

```
RP/0/RP0/CPU0:router(config)# ipv4 assembler timeout 88
```

# ipv4 mask-reply

To enable the software to respond to IPv4 Internet Control Message Protocol (ICMP) mask requests by sending ICMP mask reply messages, use the **ipv4 mask-reply** command in interface configuration mode. To restore the default, use the **no** form of this command.

**ipv4 mask-reply**
**no ipv4 mask-reply**

## Syntax Description

This command has no keywords or arguments.

## Command Default

IPv4 mask replies are not sent.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

This command enables the software to respond to IPv4 ICMP mask requests by sending ICMP mask reply messages.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |

## Examples

The following example enables the sending of ICMP mask reply messages on HundredGigEinterface 0/1/0/0:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 mask-reply
```

# ipv4 conflict-policy

To enable IP Address Repository Manager (IPARM) conflict resolution, use the **ipv4 conflict-policy** command in XR Config mode. To disable the IPARM conflict resolution, use the **no** form of the command.

**ipv4 conflict-policy** {**highest-ip** | **longest-prefix** | **static**}
**no ipv4 conflict-policy** {**highest-ip** | **longest-prefix** | **static**}

## Syntax Description

| | |
|---|---|
| **highest-ip** | Keeps the highest ip address in the conflict set. |
| **longest-prefix** | Keeps the longest prefix match in the conflict set. |
| **static** | Keeps the existing interface running across new address configurations. |

## Command Default

The precedence rule adopted is loopback > physical > other virtual interfaces. Within virtual interfaces, there is an alphabetical preference, for example, loopback1 > loopback2 > tunnel. Among physical interfaces, the lower rack or slot takes control.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Use **ipv4 conflict-policy** command to set an IPARM policy that resolves a conflict in the configured addresses. The policy tells IPARM what address to select from the addresses in conflict. The policy then forces the address in conflict to become inactive.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| ip-services | read, write |

## Examples

The following example shows how to enable the static policy for conflict resolution:

```
RP/0/RP0/CPU0:router(config)# ipv4 conflict-policy static
```

# ipv4 mtu

To set the maximum transmission unit (MTU) size of IPv4 packets sent on an interface, use the **ipv4 mtu** command in interface configuration mode.

To restore the default MTU size, use the **no** form of this command.

**ipv4 mtu** *bytes*
**no ipv4 mtu**

## Syntax Description

| | |
|---|---|
| *bytes* | MTU in bytes. Range is 68 to 65535 bytes for IPv4 packets. The maximum MTU size that can be set on an interface depends on the interface medium. |

## Command Default

If no MTU size is configured for IPv4 packets sent on an interface, the interface derives the MTU from the Layer 2 MTU.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The router punts the packets that needs fragmentation; whereas the software path drops the subscriber traffic that needs fragmentation.

The maximum MTU size that can be set on an interface depends on the interface medium. If the Layer 2 MTU is smaller than the Layer 3 MTU, the Cisco IOS XR software uses the Layer 2 MTU value for the Layer 3 MTU. Conversely, if the Layer 3 MTU is smaller than the Layer 2 MTU, the software uses Layer 3 MTU value. In other words the Cisco IOS XR software uses the lower of the two values for the MTU.

All devices on a physical medium must have the same protocol MTU to operate.



**Note** Changing the MTU value (with the **mtu** interface configuration command) can affect the IPv4 MTU value. If the current IPv4 MTU value is the same as the MTU value, and you change the MTU value, the IPv4 MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IPv4 MTU value has no effect on the value for the **mtu** command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example shows how to set the maximum IPv4 packet size for tenGigE interface 0/0/0/1 to 1500 bytes:

```
RP/0/(config)# interface tenGigE0/0/0/1
RP/0/(config-if)# ipv4 mtu 1500
```

# ipv4 unnumbered (point-to-point)

To enable IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface, use the **ipv4 unnumbered** command in an interface configuration mode. To disable this feature, use the **no** form of this command.

**ipv4 unnumbered** *interface-type interface-instance*
**no ipv4 unnumbered** *interface-type interface-instance*

## Syntax Description

| | |
|---|---|
| *interface-type* | Interface type. For more information, use the question mark (?) online help function. |
| *interface-instance* | Either a physical interface instance or a virtual interface instance as follows:<br>• Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.<br>  • *rack*: Chassis number of the rack.<br>  • *slot*: Physical slot number of the modular services card or line card.<br>  • *module*: Module number. A physical layer interface module (PLIM) is always 0.<br>  • *port*: Physical port number of the interface.<br>**Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.<br>• Virtual interface instance. Number range varies depending on interface type.<br>For more information about the syntax for the router, use the question mark (?) online help function. |

## Command Default

IPv4 processing on a point-to-point interface is disabled unless an IPv4 address is assigned explicitly to that interface.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

For release Release 4.2.0 onward, to enter the dynamic template configuration mode, run the **dynamic-template** command in the XR Config mode.

Whenever the unnumbered interface generates a packet (for example, for a routing update), it uses the address of the specified interface as the source address of the IPv4 packet. It also uses the IPv4 address of the specified

interface in determining which routing processes are sending updates over the unnumbered interface. Restrictions include the following:

- You cannot use the **ping** EXEC command to determine whether the interface is up because the interface has no address. Simple Network Management Protocol (SNMP) can be used to remotely monitor interface status.

The interface you specify by the *interface-type* and *interface-number* arguments must be enabled (listed as "up" in the **show interfaces** command display).

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example shows how the tenGigE interface 0/0/0/1 is assigned the loopback interface address 5:

```
RP/0/RP0/CPU0:router(config)# interface loopback 5
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.6.6 255.255.255.0
RP/0/RP0/CPU0:router(config)# interface tenGigE 0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5
```

# ipv6 address

To configure an IPv6 address for an interface and enable IPv6 processing on the interface using an EUI-64 interface ID in the low-order 64 bits of the address, use the **ipv6 address** command in interface configuration mode. To remove the address from the interface, use the **no** form of this command.

**ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**] [**route-tag** *route-tag value*]
**no ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**] [**route-tag** *route-tag value*]

## Syntax Description

| | |
|---|---|
| *ipv6-prefix* | The IPv6 network assigned to the interface.<br>This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. |
| */ prefix-length* | The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash (/) must precede the decimal value. |
| **eui-64** | (Optional) Specifies an interface ID in the low-order 64 bits of the IPv6 address. |
| **route-tag** | (Optional) Specifies that the configured address has a route tag to be associated with it. |
| *route-tag value* | (Optional) Value of the route tag. Range is 1 to 4294967295. |

## Command Default

No IPv6 address is defined for the interface.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |
| Release 7.6.1 | The keyword algorithm was added. |

## Usage Guidelines

If the value specified for the */ prefix-length* argument is greater than 64 bits, the prefix bits have precedence over the interface ID.

Using the **no ipv6 address** command without arguments removes all manually configured IPv6 addresses from an interface.

If the Cisco IOS XR software detects another host using one of its IPv6 addresses, it displays an error message on the console.

The route-tag feature attaches a tag to all IPv6 addresses. The tag is propagated from the Management Agents (MA) to the Address Repository Managers (RPM) to routing protocols, thus enabling the user to control the redistribution of connected routes by looking at the route tags via RPL scripts.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example assigns IPv6 address 2001:0DB8:0:1::/64 to tenGigE interface 0/0/0/1 and specifies an EUI-64 interface ID in the low-order 64 bits of the address:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 address 2001:0DB8:0:1::/64 eui-64
```

# ipv6 assembler

To configure the maximum number of packets that are allowed in assembly queues or to configure the number of seconds an assembly queue will hold before timeout , use the **ipv6 assembler** command in the appropriate configuration mode. To disable this feature, use the **no** form of this command.

**ipv6 assembler** {**max-packets** *value* | **timeout** *seconds*}
**no ipv6 assembler** {**max-packets** *value* | **timeout** *seconds*}

## Syntax Description

| | |
|---|---|
| **max-packets** | Maximum packets allowed in assembly queues. |
| **timeout** | Number of seconds an assembly queue will hold before timeout. |

## Command Default

None

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| ipv6 | read, write |

### Example

The following example shows how to configure the maximum number of packets that are allowed in assembly queues:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv6 assembler max-packets 100
```

# ipv6 conflict-policy

To enable IP Address Repository Manager (IPARM) conflict resolution, use the **ipv6 conflict-policy** command in XR Config mode mode. To disable the IPARM conflict resolution, use the **no** form of the command.

**ipv6 conflict-policy** {**highest-ip** | **longest-prefix** | **static**}
**no ipv6 conflict-policy** {**highest-ip** | **longest-prefix** | **static**}

## Syntax Description

| | |
|---|---|
| **highest-ip** | Keeps the highest IP address in the conflict set. |
| **longest-prefix** | Keeps the longest prefix match in the conflict set. |
| **static** | Keeps the existing interface running across new address configurations. |

## Command Default

Default is the lowest rack/slot if no conflict policy is configured.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| ip-services | read, write |

## Examples

The following example shows how to enable the longest prefix policy for conflict resolution:

```
RP/0/RP0/CPU0:router(config)# ipv6 conflict-policy longest-prefix
```

# ipv6 address link-local

To configure an IPv6 link-local address for an interface and enable IPv6 processing on the interface, use the **ipv6 address link-local** command in interface configuration mode. The *ipv6-address* value specified with this command overrides the link-local address that is automatically generated for the interface. To remove the address from the interface, use the **no** form of this command.

**ipv6 address** *ipv6-address* **link-local** [**route-tag** *route-tag value*]
**no ipv6 address** *ipv6-address* **link-local** [**route-tag** *route-tag value*]

## Syntax Description

| | |
|---|---|
| *ipv6-address* | The IPv6 address assigned to the interface.<br>This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. |
| **link-local** | Specifies a link-local address. The *ipv6-address* value specified with this command overrides the link-local address that is automatically generated for the interface. |
| **route-tag** | (Optional) Specifies that the configured address has a route-tag to be associated with it. |
| *route-tag value* | (Optional) Displays the route-tag value. Range is 1 to 4294967295. |

## Command Default

No IPv6 address is defined for the interface.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

If the Cisco IOS XR software detects another host using one of its IPv6 addresses, the software displays an error message on the console.

The system automatically generates a link-local address for an interface when IPv6 processing is enabled on the interface, typically when an IPv6 address is configured on the interface. To manually specify a link-local address to be used by an interface, use the **ipv6 address link-local** command.

A double colon may be used as part of the *ipv6-address* argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to assign FE80::260:3EFF:FE11:6770 as the link-local address for tenGigE interface 0/0/0/1:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local
```

# ipv6 enable

To enable IPv6 processing on an interface that has not been configured with an explicit IPv6 address, use the **ipv6 enable** command in an appropriate configuration mode. To disable IPv6 processing on an interface that has not been configured with an explicit IPv6 address, use the **no** form of this command.

**ipv6 enable**
**no ipv6 enable**

## Syntax Description

None

## Command Default

IPv6 is disabled.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **ipv6 enable** command automatically configures an IPv6 link-local unicast address on the interface while also enabling the interface for IPv6 processing. The **no ipv6 enable** command does not disable IPv6 processing on an interface that is configured with an explicit IPv6 address.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR EXEC mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to enable IPv6 processing on tenGigE interface 0/0/0/1:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 enable
```

For BNG, this example show how to enable IPv6 processing on dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp foo
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 enable
```

# ipv6 hop-limit

To configure the maximum number of hops used in router advertisements and all IPv6 packets that are originated by the router, use the **ipv6 hop-limit** command in XR Config mode mode. To return the hop limit to its default value, use the **no** form of this command.

**ipv6 hop-limit** *hops*
**no ipv6 hop-limit** *hops*

## Syntax Description

| | |
|---|---|
| *hops* | Maximum number of hops. Range is 1 to 255. |

## Command Default

*hops* : 64 hops

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to configure a maximum number of 15 hops for router advertisements and all IPv6 packets that are originated from the router:

```
RP/0/RP0/CPU0:router(config)# ipv6 hop-limit 15
```

# ipv6 icmp error-interval

To configure the interval and bucket size for IPv6 Internet Control Message Protocol (ICMP) error messages on all nodes, use the **ipv6 icmp error-interval** command in XR Config mode mode. To return the interval to its default setting, use the **no** form of this command.

**ipv6 icmp error-interval** *milliseconds* [*bucketsize*]
**no ipv6 icmp error-interval**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Time interval (in milliseconds) between tokens being placed in the bucket. Range is 0 to 2147483647. |
| *bucketsize* | (Optional) The maximum number of tokens stored in the bucket. The acceptable range is 1 to 200 with a default of 10 tokens. |

## Command Default

ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to zero.

*milliseconds* : 100 milliseconds

*bucketsize* : 10 tokens

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| | This command was introduced. |

## Usage Guidelines

Use the **ipv6 icmp error-interval** command in XR Config mode mode to limit the rate at which IPv6 ICMP error messages are sent for each node. A token bucket algorithm is used with one token representing one IPv6 ICMP error message. Tokens are placed in the virtual bucket at a specified interval until the maximum number of tokens allowed in the bucket is reached.

The *milliseconds* argument specifies the time interval between tokens being placed in the bucket. The optional *bucketsize* argument is used to define the maximum number of tokens stored in the bucket. Tokens are removed from the bucket when IPv6 ICMP error messages are sent, which means that if the *bucketsize* argument is set to 20, a rapid succession of 20 IPv6 ICMP error messages can be sent. When the bucket is empty of tokens, IPv6 ICMP error messages are not sent until a new token is placed in the bucket.

Use the **show ipv6 traffic** EXEC command to display IPv6 ICMP rate-limited counters.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
RP/0/RP0/CPU0:router(config)# ipv6 icmp error-interval 50 20
```

# ipv6 mtu

To set the maximum transmission unit (MTU) size of IPv6 packets sent on an interface, use the **ipv6 mtu** command in an appropriate configuration mode.

To restore the default MTU size, use the **no** form of this command.

**ipv6 mtu** *bytes*
**no ipv6 mtu**

## Syntax Description

| | |
|---|---|
| *bytes* | MTU in bytes. Range is 1280 to 65535 for IPv6 packets. The maximum MTU size that can be set on an interface depends on the interface medium. |

## Command Default

If no MTU size is configured for IPv6 packets sent on an interface, the interface derives the MTU from the Layer 2 MTU.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

If an IPv6 packet exceeds the MTU set for the interface, only the source router of the packet can fragment it.

The maximum MTU size that can be set on an interface depends on the interface medium. If the Layer 2 MTU is smaller than the Layer 3 MTU, the Cisco IOS XR software uses the Layer 2 MTU value for the Layer 3 MTU. Conversely, If the Layer 3 MTU is smaller than the Layer 2 MTU, the software uses Layer 3 MTU value. In other words the Cisco IOS XR software uses the lower of the two values for the MTU.

All devices on a physical medium must have the same protocol MTU to operate.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.



**Note** Changing the MTU value (with the **mtu** interface configuration command) can affect the IPv6 MTU value. If the current IPv6 MTU value is the same as the MTU value, and you change the MTU value, the IPv6 MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IPv6 MTU value has no effect on the value for the **mtu** command.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |

| Task ID | Operations |
|---|---|
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to set the maximum IPv6 packet size for tenGigE interface 0/0/0/1 to 1350 bytes:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 mtu 1350
```

For BNG, this example shows how to set the maximum IPv6 packet size to 1350 bytes in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp foo
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 mtu 1350
```

# ipv6 nd dad attempts

To configure the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on the unicast IPv6 addresses of the interface, use the **ipv6 nd dad attempts** command in an appropriate configuration mode. To return the number of messages to the default value, use the **no** form of this command.

**ipv6 nd dad attempts** *value*
**no ipv6 nd dad attempts** *value*

## Syntax Description

| | |
|---|---|
| *value* | Number of neighbor solicitation messages. Range is 0 to 600. Configuring a value of 0 disables duplicate address detection processing on the specified interface; a value of 1 configures a single transmission without follow-up transmissions. |

## Command Default

Duplicate address detection on unicast IPv6 addresses with the sending of one neighbor solicitation message is enabled. The default is one message.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Duplicate address detection verifies the uniqueness of new unicast IPv6 addresses before the addresses are assigned to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection uses neighbor solicitation messages to verify the uniqueness of unicast IPv6 addresses.

The DupAddrDetectTransmits node configuration variable (as specified in RFC 2462, *IPv6 Stateless Address Autoconfiguration*) is used to automatically determine the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on a tentative unicast IPv6 address.

The interval between the sending of duplicate address detection neighbor solicitation messages (the duplicate address detection timeout interval) is specified by the neighbor discovery-related variable RetransTimer (as specified in RFC 2461, *Neighbor Discovery for IP Version 6 [IPv6]*), which is used to determine the time between retransmissions of neighbor solicitation messages to a neighbor when the address is being resolved or when the reachability of a neighbor is being probed. This is the same management variable used to specify the interval for neighbor solicitation messages during address resolution and neighbor unreachability detection. Use the **ipv6 nd ns-interval** command to configure the interval between neighbor solicitation messages that are sent during duplicate address detection.

Duplicate address detection is suspended on interfaces that are administratively down. While an interface is administratively down, the unicast IPv6 addresses assigned to the interface are set to a pending state. Duplicate address detection is automatically restarted on an interface when the interface returns to being administratively up.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the .

**Note** An interface returning to administratively up restarts duplicate address detection for all of the unicast IPv6 addresses on the interface. While duplicate address detection is performed on the link-local address of an interface, the state for the other IPv6 addresses is still set to tentative. When duplicate address detection is completed on the link-local address, duplicate address detection is performed on the remaining IPv6 addresses.

When duplicate address detection identifies a duplicate address, the state of the address is set to duplicate and the address is not used. If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface and an error message similar to the following is issued:

```
ipv6_nd[145]: %IPV6_ND-3-ADDRESS_DUPLICATE : Duplicate address 111::1 has been detected
```

If the duplicate address is a global address of the interface, the address is not used and an error message similar to the following is issued:

```
%IPV6-4-DUPLICATE: Duplicate address 3000::4 on tenGigE&;
```

All configuration commands associated with the duplicate address remain as configured while the state of the address is set to duplicate.

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

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to set the number of consecutive neighbor solicitation messages for interface 0/2/0/1 to 1 and then display the state (tentative or duplicate) of the unicast IPv6 address configured for an interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/2/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd dad attempts 1
RP/0/RP0/CPU0:router(config-if)# Uncommitted changes found, commit them before
exiting(yes/no/cancel)? [cancel]:y

RP/0/RP0/CPU0:router# show ipv6 interface
HundredGigE/2/0/0 is Up, line protocol is Up
  IPv6 is disabled, link-local address unassigned
  No global unicast address is configured
HundredGigE/2/0/1 is Up, line protocol is Up
  IPv6 is enabled, link-local address is fe80::203:fdff:fe1b:4501
  Global unicast address(es):
    1:4::1, subnet is 1:4::/64 [DUPLICATE]
  MTU is 1514 (1500 is available to IPv6)
```

```
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
HundredGigE/2/0/2 is Shutdown, line protocol is Down
  IPv6 is enabled, link-local address is fe80::200:11ff:fe11:1111 [TENTATIVE]
  Global unicast address(es):
    111::2, subnet is 111::/64 [TENTATIVE]
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

For BNG, this example shows how to display the state (tentative or duplicate) of the unicast IPv6 address on the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd dad attempts 1
```

# ipv6 nd managed-config-flag

To set the managed address configuration flag in IPv6 router advertisements, use the **ipv6 nd managed-config-flag** command in an appropriate configuration mode. To clear the flag from IPv6 router advertisements, use the **no** form of this command.

**ipv6 nd managed-config-flag**
**no ipv6 nd managed-config-flag**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The managed address configuration flag is not set in IPv6 router advertisements.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Setting the managed address configuration flag in IPv6 router advertisements indicates to attached hosts whether they should use stateful autoconfiguration to obtain addresses. If the flag is set, the attached hosts should use stateful autoconfiguration to obtain addresses. If the flag is not set, the attached hosts should not use stateful autoconfiguration to obtain addresses.

Hosts may use stateful and stateless address autoconfiguration simultaneously.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR EXEC mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to configure the managed address configuration flag in IPv6 router advertisements on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd managed-config-flag
```

For BNG, this example shows how to configure the managed address configuration flag in IPv6 router advertisements on dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd managed-config-flag
```

# ipv6 nd ns-interval

To configure the interval between IPv6 neighbor solicitation retransmissions on an interface, use the **ipv6 nd ns-interval** command in an appropriate configuration mode. To restore the default interval, use the **no** form of this command.

**ipv6 nd ns-interval** *milliseconds*
**no ipv6 nd ns-interval**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Interval (in milliseconds) between IPv6 neighbor solicit transmissions. Range is 1000 to 3600000. |

## Command Default

0 milliseconds (unspecified) is advertised in router advertisements, and the value 1000 is used for the neighbor discovery activity of the router itself.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

This value is included in all IPv6 router advertisements sent out from this interface. Very short intervals are not recommended in normal IPv6 operation. When a nondefault value is configured, the configured time is both advertised and used by the router itself.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) configures an IPv6 neighbor solicit transmission interval of 9000 milliseconds for tenGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface tenGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ns-interval 9000
```

For BNG, this example configures an IPv6 neighbor solicit transmission interval of 9000 milliseconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd ns-interval 9000
```

# ipv6 nd other-config-flag

To set the other stateful configuration flag in IPv6 router advertisements, use the **ipv6 nd other-config-flag** command in an appropriate configuration mode. To clear the flag from IPv6 router advertisements, use the **no** form of this command.

**ipv6 nd other-config-flag**
**no ipv6 nd other-config-flag**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The other stateful configuration flag is not set in IPv6 router advertisements.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The setting of the other stateful configuration flag in IPv6 router advertisements indicates to attached hosts how they can obtain autoconfiguration information other than addresses. If the flag is set, the attached hosts should use stateful autoconfiguration to obtain the other (nonaddress) information.



**Note** If the managed address configuration flag is set using the **ipv6 nd managed-config-flag** command, then an attached host can use stateful autoconfiguration to obtain the other (nonaddress) information regardless of the setting of the other stateful configuration flag.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) configures the “other stateful configuration” flag in IPv6 router advertisements on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd other-config-flag
```

For BNG, this example configures the “other stateful configuration” flag for IPv6 router advertisements in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd other-config-flag
```

# ipv6 nd prefix

To configure how IPv6 prefixes are advertised in IPv6 router advertisements, use the **ipv6 nd prefix** command in interface configuration mode. To advertise a prefix with default parameter values, use the **no** form of this command. To prevent a prefix (or prefixes) from being advertised, use the **no- adv** keyword.

**ipv6 nd prefix** {*ipv6prefix/prefix-length* | **default** [*valid-lifetime* | **at** | **infinite** | **no-adv** | **no-autoconfig** | **off-link**]}
**no ipv6 nd prefix** {*ipv6prefix/prefix-length* | **default** [*valid-lifetime* | **at** | **infinite** | **no-adv** | **no-autoconfig** | **off-link**]}

## Syntax Description

| | |
|---|---|
| **ipv6-prefix** | The IPv6 network number to include in router advertisements.<br>This keyword must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. |
| **/prefix-length** | The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash (/) must precede the decimal value. |
| **default** | (Optional) Specifies all prefixes. |
| **valid-lifetime** | (Optional) The amount of time (in seconds) that the specified IPv6 prefix is advertised as being valid. The range of values is 0 to 4294967295 seconds. |
| **at** | (Optional) The date and time at which the lifetime and preference expire. The prefix is valid until this specified date and time are reached. Dates are expressed in the form *date-valid-expire month-valid-expire hh:mm-valid-expire date-prefer-expire month-prefer-expire hh:mm-prefer-expire.* |
| **infinite** | (Optional) The valid lifetime does not expire. |
| **no-adv** | (Optional) The prefix is not advertised. |
| **no-autoconfig** | (Optional) Indicates to hosts on the local link that the specified prefix cannot be used for IPv6 autoconfiguration. |
| **off-link** | (Optional) Indicates that the specified prefix is assigned to the link. Nodes sending traffic to such addresses that contain the specified prefix consider the destination to be locally reachable on the link. This prefix should not be used for *onlink* determination. |

## Command Default

All prefixes configured on interfaces that originate IPv6 router advertisements are advertised with a valid lifetime of 2592000 seconds (30 days) and a preferred lifetime of 604800 seconds (7 days), and with both the “onlink” and “autoconfig” flags set.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

This command allows control over the individual parameters per prefix, including whether or not the prefix should be advertised.

To control how prefixes are advertised, use the **ipv6 nd prefix** command. By default, prefixes configured as addresses on an interface using the **ipv6 address** command are advertised with default values. If you configure prefixes for advertisement using the **ipv6 nd prefix** command, only the specified prefixes are advertised with the configured values, all other prefixes are advertised with default values.

The default keyword can be used to set default parameters for all prefixes.

A date can be set to specify the expiration of a prefix. The valid and preferred lifetimes are counted down in real time. When the expiration date is reached, the prefix is no longer advertised.

When onlink is “on” (by default), the specified prefix is assigned to the link. Nodes sending traffic to such addresses that contain the specified prefix consider the destination to be locally reachable on the link.

When autoconfig is “on” (by default), it indicates to hosts on the local link that the specified prefix can be used for IPv6 autoconfiguration.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example includes the IPv6 prefix 2001:0DB8::/35 in router advertisements sent out HundredGigE interface 0/1/0/0 with a valid lifetime of 1000 seconds and a preferred lifetime of 900 seconds:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv6 nd prefix 2001:0DB8::/35 1000 900
```

# ipv6 nd ra-interval

To configure the interval between IPv6 router advertisement transmissions on an interface, use the **ipv6 nd ra-interval** command in an appropriate configuration mode. To restore the default interval, use the **no** form of this command.

**ipv6 nd ra-interval** *seconds*
**no ipv6 nd ra-interval** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | The interval (in seconds) between IPv6 router advertisement transmissions. |

## Command Default

*seconds* : 200 seconds

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

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

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) configures an IPv6 router advertisement interval of 201 seconds on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ra-interval 201
```

For BNG, this example configures an IPv6 router advertisement interval of 201 seconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd ra-interval 201
```

# ipv6 nd ra-lifetime

To configure the router lifetime value in IPv6 router advertisements on an interface, use the **ipv6 nd ra-lifetime** command in an appropriate configuration mode. To restore the default lifetime, use the **no** form of this command.

**ipv6 nd ra-lifetime** *seconds*
**no ipv6 nd ra-lifetime**

## Syntax Description

| | |
|---|---|
| *seconds* | The validity (in seconds) of this router as a default router on this interface. |

## Command Default

*seconds* : 1800 seconds

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Usage Guidelines

The router lifetime value is included in all IPv6 router advertisements sent out the interface. The value indicates the usefulness of the router as a default router on this interface. Setting the value to 0 indicates that the router should not be considered a default router on this interface. The router lifetime value can be set to a nonzero value to indicate that it should be considered a default router on this interface. The nonzero value for the router lifetime value should not be less than the router advertisement interval.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) configures an IPv6 router advertisement lifetime of 1801 seconds on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ra-lifetime 1801
```

For BNG, this example configures an IPv6 router advertisement lifetime of 1801 seconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd ra-lifetime 1801
```

# ipv6 nd reachable-time

To configure the amount of time that a remote IPv6 node is considered reachable after some reachability confirmation event has occurred, use the **ipv6 nd reachable-time** command in an appropriate configuration mode. To restore the default time, use the **no** form of this command.

**ipv6 nd reachable-time** *milliseconds*
**no ipv6 nd reachable-time**

## Syntax Description

| | |
|---|---|
| *milliseconds* | The amount of time (in milliseconds) that a remote IPv6 node is considered reachable. The range is from 0 to 3600000. |

## Command Default

0 milliseconds (unspecified) is advertised in router advertisements and 30000 (30 seconds) is used for the neighbor discovery activity of the router itself.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

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

The configured time is included in all router advertisements sent out of an interface so that nodes on the same link use the same time value. A value of 0 indicates that the configured time is unspecified by this router.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to configure an IPv6 reachable time of 1,700,000 milliseconds for HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
```

```
RP/0/RP0/CPU0:router(config-if)# ipv6 nd reachable-time 1700000
```

For BNG, this example shows how to configure an IPv6 reachable time of 1,700,000 milliseconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd reachable-time 1700000
```

# ipv6 nd redirects

To send Internet Control Message Protocol (ICMP) redirect messages, use the **ipv6 nd redirects** command in interface configuration mode. To restore the system default, use the **no** form of this command.

**ipv6 nd redirects**
**no ipv6 nd redirects**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The default value is disabled.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

This command has no keywords or arguments.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to redirect IPv6 nd-directed broadcasts on HundredGigE interface 0/2/0/2:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/2/0/2
RP/0/RP0/CPU0:router(config-if)# ipv6 nd redirects
```

# ipv6 nd scavenge-timeout

To set the lifetime for neighbor entries in the stale state, use the **ipv6 nd scavenge-timeout** command in XR Config mode mode. To disable this feature, use the **no** form of this command.

**ipv6 nd scavenge-timeout** *seconds*
**no ipv6 nd scavenge-timeout** *seconds*

## Syntax Description

| | |
|---|---|
| seconds | RA lifetime in seconds. The range is from 0 to 43200. |

## Command Default

None

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

When the scavenge-timer for a neighbor entry expires, the entry is cleared.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to set the lifetime for the neighbor entry:

```
RP/0/RP0/CPU0:router(config)# ipv6 nd scavenge-timeout 3000
```

# ipv6 nd suppress-ra

To suppress IPv6 router advertisement transmissions on a LAN interface, use the **ipv6 nd suppress-ra** command in an appropriate configuration mode. To reenable the sending of IPv6 router advertisement transmissions on a LAN interface, use the **no** form of this command.

**ipv6 nd suppress-ra**
**no ipv6 nd suppress-ra**

## Syntax Description

This command has no keywords or arguments.

## Command Default

IPv6 router advertisements are automatically sent on other types of interlaces if IPv6 unicast routing is enabled on the interfaces. IPv6 router advertisements are not sent on other types of interfaces.

## Command Modes

Interface configuration (not applicable for BNG)

Dynamic template configuration (for BNG)

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Use the **no ipv6 nd suppress-ra** command to enable the sending of IPv6 router advertisement transmissions on non-LAN interface types (for example, serial or tunnel interfaces).

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example (not applicable for BNG) shows how to suppress IPv6 router advertisements on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd suppress-ra
```

For BNG, this example shows how to suppress IPv6 router advertisements in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd suppress-ra
```

# ipv6 neighbor

To configure a static entry in the IPv6 neighbor discovery cache, use the **ipv6 neighbor** command in XR Config mode mode. To remove a static IPv6 entry from the IPv6 neighbors discovery cache, use the **no** form of this command.

**ipv6 neighbor** *ipv6-address interface-type interface-instance hardware-address*
**no ipv6 neighbor** *ipv6-address interface-type interface-instance hardware-address*

## Syntax Description

| | |
|---|---|
| *ipv6-address* | The IPv6 address that corresponds to the local data-link address.<br>This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. |
| *interface-type* | Interface type. For more information, use the question mark (?) online help function. |
| *interface-instance* | Either a physical interface instance or a virtual interface instance as follows:<br>• Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.<br>  • *rack*: Chassis number of the rack.<br>  • *slot*: Physical slot number of the modular services card or line card.<br>  • *module*: Module number. A physical layer interface module (PLIM) is always 0.<br>  • *port*: Physical port number of the interface.<br>**Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.<br>• Virtual interface instance. Number range varies depending on interface type.<br>For more information about the syntax for the router, use the question mark (?) online help function. |
| *hardware-address* | The local data-link address (a 48-bit address). |

## Command Default

Static entries are not configured in the IPv6 neighbor discovery cache.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **ipv6 neighbor** command is similar to the **arp** (global) command.

If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry.

Use the **show ipv6 neighbors** command to display static entries in the IPv6 neighbors discovery cache. A static entry in the IPv6 neighbor discovery cache has one state: reach (reachable)—The interface for this entry is up. If the interface for the entry is down, the **show ipv6 neighbors** command does not show the entry.

**Note** Reachability detection is not applied to static entries in the IPv6 neighbor discovery cache; therefore, the descriptions for the reach (reachable) state are different for dynamic and static cache entries. See the **show ipv6 neighbors** command for a description of the reach (reachable) state for dynamic cache entries.

The **clear ipv6 neighbors** command deletes all entries in the IPv6 neighbors discovery cache, except static entries. The **no ipv6 neighbor** command deletes a specified static entry from the neighbor discovery cache; the command does not remove dynamic entries—learned from the IPv6 neighbor discovery process—from the cache. Disabling IPv6 on an interface by using the **no ipv6 enable** or the **no ipv6 unnumbered** command deletes all IPv6 neighbor discovery cache entries configured for that interface, except static entries (the state of the entry changes to reach [reachable]).

Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.

**Note** Static entries for IPv6 neighbors can be configured only on IPv6-enabled LAN and ATM LAN Emulation interfaces.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to configure a static entry in the IPv6 neighbor discovery cache for a neighbor with the IPv6 address 2001:0DB8::45A and link-layer address 0002.7D1A.9472 on ethernet interface 0/RSP0/CPU0:

```
RP/0/RP0/CPU0:router(config)# ipv6 neighbor 2001:0DB8::45A 0002.7D1A.9472
```

# ipv6 path-mtu enable

To enable the command to configure path maximum transmission unit (MTU) discovery of IPv6 packets, use the **ipv6 path-mtu enable** command in the XR Config mode.

**ipv6 path-mtu enable**

## Command Default

None.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example shows how to enable path MTU discovery command of IPv6 packets:

```
RP/0/RP0/CPU0:router(config)#  ipv6 path-mtu enable
```

# ipv6 path-mtu timeout

To set the maximum transmission unit (MTU) timeout value of IPv6 packets, use the **ipv6 path-mtu timeout** command in the XR Config mode.

**ipv6 path-mtu timeout** *minutes*

**Syntax Description**

| | |
|---|---|
| *minutes* | MTU timeout in minutes. Range is 1 to 15 minutes. Default timeout value is 10 minutes. |

**Command Default**

None.

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

**Usage Guidelines**

Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.

**Task ID**

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

**Examples**

This example shows how to set path MTU timeout of IPv6 packets:

```
RP/0/RP0/CPU0:router(config)#  ipv6 path-mtu timeout 15
```

# ipv6 source-route

To enable processing of the IPv6 type source (type 0) routing header, use the **ipv6 source-route** command in XR EXEC mode mode. To disable the processing of this IPv6 extension header, use the **no** form of this command.

**ipv6 source-route**
**no ipv6 source-route**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The **no** version of the **ipv6 source-route** command is the default.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **no ipv6 source-route** command (which is the default) prevents hosts from performing source routing using your routers. When the **no ipv6 source-route** command is configured and the router receives a packet with a type 0 source routing header, the router drops the packet and sends an IPv6 ICMP error message back to the source and logs an appropriate debug message.

## Task ID

| Task ID | Operation |
|---|---|
| network | read, write |
| ipv6 | read, write |

### Example

The following example shows how to allow the processing of any IPv6 datagrams containing a source-route header option:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv6 source-route
RP/0/RP0/CPU0:router(config)#
```

# ipv6 tcp-mss-adjust

To enable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU so that traffic fragmentation is prevented when a session is established for IPv6 packets, use the **ipv6 tcp-mss-adjust** command in the interface configuration submode. To disable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU, use the **no** form of this command.

**ipv6 tcp-mss-adjust enable**
**no ipv6 tcp-mss-adjust enable**

## Syntax Description

| | |
|---|---|
| **enable** | Enables Maximum Segment Size (MSS) adjustment for tcp flows on the interface.. |

## Command Default

None

## Command Modes

Interface Configuration

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

This command has no keywords or arguments.

## Task ID

| Task ID | Operation |
|---|---|
| mpls-te | read, write |
| ipv6 | read, write |

### Example

This example shows how to enable the transit traffic of TCP flows for IPv6 packets using the **ipv6 tcp-mss-adjust** command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredEthernet 0/0/0/4.20
RP/0/RP0/CPU0:router(config-if)# ipv6 tcp-mss-adjust enable
```

# ipv6 virtual address

To define an IPv6 virtual address for a network of management Ethernet interfaces, use the **ipv6 virtual address** command in XR Config mode. To remove an IPv6 virtual address from the configuration, use the **no** form of this command.

**ipv6 virtual address** {[**vrf** *vrf-name*]*ipv6-address/prefix-length* | **use-as-src-addr**}
**no ipv6 virtual address** {[**vrf** *vrf-name*]*ipv6-address/prefix-length* | **use-as-src-addr**}

## Syntax Description

| | |
|---|---|
| **vrf** *vrf-name* | (Optional) Configures the virtual address on a per VPN routing and forwarding (VRF) basis for the management interfaces The *vrf-name* argument specifies the name of the VRF. |
| *ipv6 address* | The virtual IPv6 address to be used. |
| *prefix-length* | The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. |
| **use-as-src-addr** | Enables the virtual address to be used as the default SRC address on sourced packets. |

## Command Default

No IPv6 virtual address is defined for the configuration.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Configuring an IPv6 virtual address enables you to access the router from a single virtual address with a management network. An IPv6 virtual address persists across route processor (RP) failover situations.

Configuring an IPv6 virtual address enables you to access a dual RP router from a single address without prior knowledge of which RP is active. An IPv6 virtual address persists across RP failovers. For this to happen, the virtual IPv6 address must share a common IPv6 subnet with a Management Ethernet interface on both RPs.

If you disable the **ipv6 virtual address** command with the **vrf** keyword, the virtual IP address is unconfigured for the corresponding VRF or for the default if no VRF is specified. This results in the removal of the entry for the virtual IP address in the VRF table and in the ARP cache.

The default VRF is chosen when no VRF is specified. The virtual IP address is activated on a management interface that is attached to a default VRF.

The **use-as-src-addr** keyword eliminates the need for configuring a loopback interface as the source interface (that is, update source) for management applications. When an update source is not configured, management applications allow the transport processes (TCP, UDP, raw_ip) to pick a suitable source address. The transport processes, in turn, consult the FIB to do so. If a Management Ethernet's IP address is picked as the source address and if the **use-as-src-addr keyword** is configured, then the transport processes replace the Management Ethernet's IP address with a relevant virtual IP address. This functionality works across RP switchovers.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |

## Examples

The following example shows how to define an IPv6 virtual address:

```
RP/0/RP0/CPU0:router(config)# ipv6 virtual address 0:0:0:7272::72/64
```

The following example shows how to configure the virtual IP addresses for management interfaces on a per VRF basis:

```
RP/0/RP0/CPU0:router(config)# ipv6 virtual address vrf ppp 0:0:0:7272::72/64
```

# show arm conflicts

To display IPv4 or IPv6 address conflict information identified by the Address Repository Manager (ARM), use the **show arm conflicts** command in XR EXEC mode.

**show arm** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **conflicts** [**address** | **override** | **unnumbered**]

## Syntax Description

| | |
|---|---|
| **ipv4** | Displays IPv4 address conflicts. |
| **ipv6** | Displays IPv6 address conflicts. |
| **vrf** | (Optional) Displays VPN routing and forwarding (VRF) instance information. Available for IPv4 only. |
| *vrf-name* | (Optional) Name of a VRF. |
| **address** | (Optional) Displays address conflict information. |
| **override** | (Optional) Displays address conflict override information. |
| **unnumbered** | (Optional) Displays unnumbered interface conflict information. |

## Command Default

None

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Use the **show arm conflicts** command to display information about IPv4 or IPv6 address conflicts. You can use address conflict information to identify misconfigured IPv4 or IPv6 addresses.

Conflict information is displayed for interfaces that are forced down and for interfaces that are up.

Issuing the **show arm conflicts** command without specifying any optional keywords displays the output generated from both the **address** and **unnumbered** keywords.

## Task ID

| Task ID | Operations |
|---|---|
| network | read |

## Examples

The following sample output is from the **show arm ipv4 conflicts** command:

```
RP/0/RP0/CPU0:router# show arm ipv4 conflicts

F Forced down
| Down interface & addr                  Up interface & addr

F Lo2 10.1.1.2/24                         Lo1 10.1.1.1/24

Forced down interface          Up interface
```

```
tu2->tu1                        tu1->Lo1
```

The following is sample output from the **show arm ipv4 conflicts** command with the **address** keyword:

```
RP/0/RP0/CPU0:router# show arm ipv4 conflicts address

F Forced down
| Down interface & addr                   Up interface & addr

F Lo2 10.1.1.2/24                          Lo1 10.1.1.1/24
```

The following is sample output from the **show arm ipv4 conflicts** command with the **unnumbered** keyword:

```
RP/0/RP0/CPU0:router# show arm ipv4 conflicts unnumbered

Forced down interface          Up interface                   VRF
tu2->tu1                       tu1->Lo1
```

This table describes the significant fields shown in the display.

***Table 1: show arm conflicts Command Field Descriptions***

| Field | Description |
|---|---|
| Forced down | Legend defining a symbol that may appear in the output for this command. |
| Down interface & addr | Forced down interface name, type, and address. |
| Up interface & addr | List of interfaces that are up. |
| Forced down interface | Unnumbered interfaces that are in conflict and forced down. |
| Up interface | Unnumbered interfaces that are in conflict and are up. |

# show arm registrations producers

To display producer registration information for the Address Repository Manager (ARM), use the **show arm registrations producers** command in XR EXEC mode.

**show arm** {**ipv4** | **ipv6**} **registrations producers**

## Syntax Description

| | |
|---|---|
| **ipv4** | Displays IPv4 producer registration information. |
| **ipv6** | Displays IPv6 producer registration information. |

## Command Default

None

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

Use the **show arm registrations producers** command to display information on producers of IP ARM registrations. Registration information is displayed with the ID.

## Task ID

| Task ID | Operations |
|---|---|
| network | read |

## Examples

The following is sample output from the **show arm registrations producers** command:

```
RP/0/RP0/CPU0:router# show arm ipv4 registrations producers

Id    Node            Producer Id   IPC Version Connected?
0     0/0/0           ipv4_io       1.1         Y
4     0/1/0           ipv4_io       1.1         Y
3     0/2/0           ipv4_io       1.1         Y
2     0/4/0           ipv4_io       1.1         Y
1     0/6/0           ipv4_io       1.1         Y
```

This table describes the significant fields shown in the display.

***Table 2: show arm registrations producers Command Field Descriptions***

| Field | Description |
|---|---|
| Id | An identifier used by the IP Address ARM (IP ARM) to keep track of the producer of the IP address. |
| Node | The physical node (RP/LC CPU) where the producer is running. |
| Producer Id | The string used by the producer when registering with IP ARM. |

| Field | Description |
|---|---|
| IPC Version | Version of the apis used by the producer to communicate with IP ARM. |
| Connected? | Status of whether the producer is connected or not. |

# show arm router-ids

To display the router identification information with virtual routing and forwarding table information for the Address Repository Manager (ARM), use the **show arm router-ids** command in XR EXEC mode.

**show arm** [**ipv4**] **router-ids**

**Syntax Description**

| | |
|---|---|
| **ipv4** | (Optional) Displays IPv4 router information. |

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

**Usage Guidelines**

Use the **show arm router-ids** command with the **ipv4** keyword to display the selected router ID information for the router.

**Task ID**

| Task ID | Operations |
|---|---|
| network | read |

**Examples**

The following is sample output from the **show arm router-ids** command:

```
RP/0/RP0/CPU0:router# show arm router-ids

Router-ID         Interface

10.10.10.10       Loopback0
```

This table describes the significant fields shown in the display.

***Table 3: show arm router-ids Command Field Descriptions***

| Field | Description |
|---|---|
| Router-ID | Router identification. |
| Interface | Interface identification. |

# show arm summary

To display summary information for the IP Address Repository Manager (ARM), use the **show arm summary** command in XR EXEC mode.

**show arm** {**ipv4** | **ipv6**} **summary**

**Syntax Description**

| | |
|---|---|
| **ipv4** | Displays IPv4 summary information. |
| **ipv6** | Displays IPv6 summary information. |

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

**Usage Guidelines**

Use the **show arm summary** command to display a summary of the number of producers, address conflicts, and unnumbered interface conflicts in the router.

**Task ID**

| Task ID | Operations |
|---|---|
| network | read |

**Examples**

The following is sample output from the **show arm summary** command:

```
Router# show arm ipv4 summary

IPv4 Producers                          :          1
IPv4 address conflicts                  :          0
IPv4 unnumbered interface conflicts     :          0
IPv4 VRF known                          :          0
IPv4 DB Master version                  : 0x00000000
```

This table describes the significant fields shown in the display.

***Table 4: show arm summary Command Field Descriptions***

| Field | Description |
|---|---|
| IPv4 Producers | Number of IPv4 producers on the router. |
| IPv4 address conflicts | Number of IPv4 address conflicts on the router. |
| IPv4 unnumbered interface conflicts | Number of IPv4 conflicts on unnumbered interfaces. |
| IPv4 DB Master version | IPv4 DB Master version |

# show ipv4 interface

To display the usability status of interfaces configured for IPv4, use the **show ipv4 interface** command in the XR EXEC mode.

**show ipv4 interface** [*type interface-path-id* | **brief** | **summary**]

## Syntax Description

| | |
|---|---|
| *type* | Interface type. For more information, use the question mark (?) online help function. |
| *interface-path-id* | Either a physical interface instance or a virtual interface instance as follows:<br>• Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.<br>  • *rack*: Chassis number of the rack.<br>  • *slot*: Physical slot number of the modular services card or line card.<br>  • *module*: Module number. A physical layer interface module (PLIM) is always 0.<br>  • *port*: Physical port number of the interface.<br>**Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.<br>• Virtual interface instance. Number range varies depending on interface type.<br>For more information about the syntax for the router, use the question mark (?) online help function. |
| **brief** | (Optional) Displays the primary IPv4 addresses configured on the router's interfaces and their protocol and line states. |
| **summary** | (Optional) Displays the number of interfaces on the router that are assigned, unassigned, or unnumbered. |

## Command Default

If VRF is not specified, the software displays the default VRF.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **show ipv4 interface** command provides output similar to the **show ipv6 interface** command, except that it is IPv4-specific.

The interface name will be displayed only if the name belongs to the VRF instance. If the *vrf-name* is not specified then the interface instance will be displayed only if the interface belongs to the default VRF.

## Task ID

| Task ID | Operations |
|---|---|
| ipv4 | read |
| network | read |

## Examples

This is the sample output of the **show ipv4 interface** command:

```
RP/0/RP0/CPU0:router# show ipv4 interface

  Loopback0 is Up, line protocol is Up
  Internet address is 10.0.0.1/8

  Secondary address 10.0.0.2/8
  MTU is 1514 (1514 is available to IP)
  Multicast reserved groups joined: 10.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  tenGigE/0/0/0/0 is Up, line protocol is Up
  Internet address is 10.25.58.1/16
  MTU is 1514 (1500 is available to IP)
  Multicast reserved groups joined: 10.0.224.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  tenGigE/0/0/0/0 is Shutdown, line protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet protocol processing disabled
```

This table describes the significant fields shown in the display.

***Table 5: show ipv4 interface Command Field Descriptions***

| Field | Description |
|---|---|
| Loopback0 is Up | If the interface hardware is usable, the interface is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up. |
| line protocol is Up | If the interface can provide two-way communication, the line protocol is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up. |
| Internet address | IPv4 Internet address and subnet mask of the interface. |

| Field | Description |
|---|---|
| Secondary address | Displays a secondary address, if one has been set. |
| MTU | Displays the IPv4 MTU[1] value set on the interface. |
| Multicast reserved groups joined | Indicates the multicast groups this interface belongs to. |
| Directed broadcast forwarding | Indicates whether directed broadcast forwarding is enabled or disabled. |
| Outgoing access list | Indicates whether the interface has an outgoing access list set. |
| Inbound access list | Indicates whether the interface has an incoming access list set. |
| Proxy ARP | Indicates whether proxy ARP[2] is enabled or disabled on an interface. |
| ICMP redirects | Specifies whether ICMPv4[3] redirects are sent on this interface. |
| ICMP unreachables | Specifies whether unreachable messages are sent on this interface. |
| Internet protocol processing disabled | Indicates an IPv4 address has not been configured on the interface. |

1 MTU = maximum transmission unit
2 ARP = Address Resolution Protocoladdress resolution protocol
3 ICMPv4 = Internet Control Message Protocol internet control message protocol version 4

# show ipv4 traffic

To display the IPv4 traffic statistics, use the **show ipv4 traffic** command in the XR EXEC mode.

**show ipv4 traffic** [**brief**]

**Syntax Description**

| | |
|---|---|
| **brief** | (Optional) Displays only IPv4 and Internet Control Message Protocol version 4 (ICMPv4) traffic. |

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

**Usage Guidelines**

The **show ipv4 traffic** command provides output similar to the **show ipv6 traffic** command, except that it is IPv4-specific.

**Task ID**

| Task ID | Operations |
|---|---|
| ipv4 | read |
| network | read |

**Examples**

This is the sample output of the **show ipv4 traffic** command:

```
RP/0/RP0/CPU0:router# show ipv4 traffic

IP statistics:
  Rcvd:  16372 total, 16372 local destination
         0 format errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad source, 0 bad header
         0 with options, 0 bad, 0 unknown
  Opts:  0 end, 0 nop, 0 basic security, 0 extended security
         0 strict source rt, 0 loose source rt, 0 record rt
         0 stream ID, 0 timestamp, 0 alert, 0 cipso
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragment count
  Bcast: 0 sent, 0 received
  Mcast: 0 sent, 0 received
   Drop: 0 encapsulation failed, 0 no route, 0 too big, 0 sanity address check
   Sent: 16372 total

ICMP statistics:
  Sent: 0 admin unreachable, 0 network unreachable
        0 host unreachable, 0 protocol unreachable
        0 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        5 echo request, 0 echo reply
```

```
        0 mask request, 0 mask reply
        0 parameter error, 0 redirects
        5 total
  Rcvd: 0 admin unreachable, 0 network unreachable
        2 host unreachable, 0 protocol unreachable
        0 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        0 echo request, 5 echo reply
        0 mask request, 0 mask reply
        0 redirect, 0 parameter error
        0 source quench, 0 timestamp, 0 timestamp reply
        0 router advertisement, 0 router solicitation
        7 total, 0 checksum errors, 0 unknown

UDP statistics:
        16365 packets input, 16367 packets output
        0 checksum errors, 0 no port
        0 forwarded broadcasts

TCP statistics:
        0 packets input, 0 packets output
        0 checksum errors, 0 no port
```

This table describes the significant fields shown in the display.

***Table 6: show ipv4 traffic Command Field Descriptions***

| Field | Description |
|---|---|
| bad hop count | Occurs when a packet is discarded because its TTL[4] field was decremented to zero. |
| encapsulation failed | Usually indicates that the router had no ARP request entry and therefore did not send a datagram. |
| format errors | Indicates a gross error in the packet format, such as an impossible Internet header length. |
| IP statistics Rcvd total | Indicates the total number of local destination and other packets received in the software plane. It does not account for the IP packets forwarded or discarded in hardware. |
| no route | Counted when the Cisco IOS XR software discards a datagram it did not know how to route. |

[4] TTL = time-to-live

# show ipv6 interface

To display the usability status of interfaces configured for IPv6, use the **show ipv6 interface** command in the XR EXEC mode.

**show ipv6 interface** [**summary** | [*type interface-path-id*][**brief** [**link-local** | **global**]]]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Interface type. For more information, use the question mark (?) online help function. |
| *interface-path-id* | (Optional) Either a physical interface instance or a virtual interface instance as follows:<br>• Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.<br>  • *rack*: Chassis number of the rack.<br>  • *slot*: Physical slot number of the modular services card or line card.<br>  • *module*: Module number. A physical layer interface module (PLIM) is always 0.<br>  • *port*: Physical port number of the interface.<br>**Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.<br>• Virtual interface instance. Number range varies depending on interface type.<br>For more information about the syntax for the router, use the question mark (?) online help function. |
| **brief** | (Optional) Displays the primary IPv6 addresses configured on the router interfaces and their protocol and line states. |
| **link-local** | (Optional) Displays the link local IPv6 address. |
| **global** | (Optional) Displays the global IPv6 address. |
| **summary** | (Optional) Displays the number of interfaces on the router that are assigned, unassigned, or unnumbered. |

## Command Default

None

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **show ipv6 interface** command provides output similar to the **show ipv4 interface** command, except that it is IPv6-specific.

Use the **link-local** or **global** keywords along with the **brief** keyword to view the link local or global IPv6 addresses.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read |

## Examples

This is the sample output of the **show ipv6 interface** command:

```
RP/0/RP0/CPU0:router# show ipv6 interface

  TenGigE0/0/0/0 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::c672:95ff:fea6:9324
  Global unicast address(es):
    200::1, subnet is 200::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ffa6:9324 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 2
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
```

This table describes the significant fields shown in the display.

***Table 7: show ipv6 interface Command Field Descriptions***

| Field | Description |
|---|---|
| tenGigE0/0/0/2 is Up | Indicates whether the interface hardware is currently active (whether line signal is present) and whether it has been taken down by an administrator. If the interface hardware is usable, the interface is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up. |

| Field | Description |
|---|---|
| line protocol is Up (or down) | Indicates whether the software processes that handle the line protocol consider the line usable (that is, whether keepalives are successful). If the interface can provide two-way communication, the line protocol is marked "Up." For an interface to be usable, both the interface hardware and line protocol must be up. |
| IPv6 is enabled, stalled, disabled (stalled and disabled are not shown in sample output) | Indicates that IPv6 is enabled, stalled, or disabled on the interface. If IPv6 is enabled, the interface is marked "enabled." If duplicate address detection processing identified the link-local address of the interface as being a duplicate address, the processing of IPv6 packets is disabled on the interface and the interface is marked "stalled." If IPv6 is not enabled, the interface is marked "disabled." |
| link-local address | Displays the link-local address assigned to the interface. |
| TENTATIVE | The state of the address in relation to duplicate address detection. States can be any of the following:<br>• duplicate—The address is not unique and is not being used. If the duplicate address is the link-local address of an interface, the processing of IPv6 packets is disabled on that interface.<br>• tentative—Duplicate address detection is either pending or under way on this interface.<br>**Note**<br>If an address does not have one of these states (the state for the address is blank), the address is unique and is being used. |
| Global unicast addresses | Displays the global unicast addresses assigned to the interface. |
| ICMP redirects | State of Internet Control Message Protocol (ICMP) IPv6 redirect messages on the interface (the sending of the messages is enabled or disabled). |
| ND DAD | State of duplicate address detection on the interface (enabled or disabled). |
| number of DAD attempts | Number of consecutive neighbor solicitation messages that are sent on the interface while duplicate address detection is performed. |
| ND reachable time | Displays the neighbor discovery reachable time (in milliseconds) assigned to this interface. |

This is the sample output of the **show ipv6 interface brief link-local** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface brief link-local

Interface                  IPv6-Address                    Status     Protocol
tenGigE0/0/0/0    fe80::fe:8ff:fecb:26c5        Up         Up
tenGigE0/0/0/1    fe80::4f:88ff:fea0:8c9d       Up         Up
tenGigE0/0/0/3    unassigned                    Shutdown   Down
tenGigE0/0/0/4    unassigned                    Shutdown   Down
```

This is the sample output of the **show ipv6 interface brief global** command:

```
RP/0/#show ipv6 interface brief global

Interface       IPv6-Address                  Status     Protocol
tenGigE0/0/0/0    2001:db8::1                   Up         Up
tenGigE0/0/0/1    2001:db8::2                   Up         Up
tenGigE0/0/0/3    unassigned                    Shutdown   Down
tenGigE0/0/0/4    unassigned                    Shutdown   Down
```

This is the sample output of the **show ipv6 interface** *type interface-path-id* **brief link-local** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface tenGigE 0/0/0/0 brief link-local

Interface                  IPv6-Address                  Status     Protocol
tenGigE0/0/0/0    fe80::fe:8ff:fecb:26c5        Up         Up
```

This is the sample output of the **show ipv6 interface** *type interface-path-id* **brief global** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface tenGigE 0/0/0/0 brief global

Interface                  IPv6-Address                  Status     Protocol
tenGigE0/0/0/0    2001:db8::1                   Up         Up
```

# show ipv6 neighbors

To display the IPv6 neighbor discovery cache information, use the **show ipv6 neighbors** command in the XR EXEC mode.

**show ipv6 neighbors** [*type interface-path-id* | **location** *node-id*]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Interface type. For more information, use the question mark (?) online help function. |
| *interface-path-id* | (Optional) Physical interface instance or a virtual interface.<br>**Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>For more information about the syntax for the router, use the question mark (?) online help function. |
| **location** *node-id* | (Optional) Designates a node. The *node-id* argument is entered in the *rack/slot/module* notation. |

## Command Default

All IPv6 neighbor discovery cache information is displayed.

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

When the *interface-type* and *interface-number* arguments are not specified, cache information for all IPv6 neighbors is displayed. Specifying the *interface-type* and *interface-number* arguments displays only cache information about the specified interface.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read |

## Examples

This is the sample output of the **show ipv6 neighbors** command when entered with an IPv6 address:

```
RP/0/RP0/CPU0:router# show ipv6 neighbors 2000:0:0:4::2

IPv6 Address                              Age Link-layer Addr State Interface
Location
200::2                                    166  6c9c.ed28.8b74 REACH Te0/0/0/0
0/RP0/CPU0
fe80::6e9c:edff:fe28:8b74                 164  6c9c.ed28.8b74 REACH Te0/0/0/0
0/RP0/CPU0
[Mcast adjacency]                           - 0000.0000.0000 REACH Te0/0/0/0
0/RP0/CPU0
```

```
[Mcast adjacency]                           - 0000.0000.0000 REACH Te0/0/0/4
0/RP0/CPU0
```

This is the sample output of the **show ipv6 neighbors** command when entered with a location:

```
RP/0/RP0/CPU0:router# show ipv6 neighbors  location 0/2/CPU0

IPv6 Address    Age Link-layer Addr State Interface     Location
2001:3::2       119  0013.9400.0002 REACH BE3           0/2/CPU0
2001:3::3       179  0013.9400.0003 DELAY BE3           0/2/CPU0
2001:3::4       166  0013.9400.0004 REACH BE3           0/2/CPU0
2001:3::5       78   0013.9400.0005 REACH BE3           0/2/CPU0
2001:3::6       19   0013.9400.0006 REACH BE3           0/2/CPU0
2001:3::7       173  0013.9400.0007 REACH BE3           0/2/CPU0
2001:3::8       140  0013.9400.0008 REACH BE3           0/2/CPU0
2001:3::9       163  0013.9400.0009 REACH BE3           0/2/CPU0
2001:3::a       40   0013.9400.000a REACH BE3           0/2/CPU0
2001:3::b       90   0013.9400.000b REACH BE3           0/2/CPU0
2001:3::c       35   0013.9400.000c REACH BE3           0/2/CPU0
2001:3::d       114  0013.9400.000d REACH BE3           0/2/CPU0
2001:3::e       117  0013.9400.000e REACH BE3           0/2/CPU0
2001:3::f       157  0013.9400.000f REACH BE3           0/2/CPU0
2001:3::10      9    0013.9400.0010 REACH BE3           0/2/CPU0
2001:3::11      120  0013.9400.0011 REACH BE3           0/2/CPU0
2001:3::12      87   0013.9400.0012 REACH BE3           0/2/CPU0
2001:3::13      180  0013.9400.0013 DELAY BE3           0/2/CPU0
2001:3::14      103  0013.9400.0014 REACH BE3           0/2/CPU0
2001:3::15      132  0013.9400.0015 REACH BE3           0/2/CPU0
2001:3::16      33   0013.9400.0016 REACH BE3           0/2/CPU0
2001:3::17      150  0013.9400.0017 REACH BE3           0/2/CPU0
2001:3::18      117  0013.9400.0018 REACH BE3           0/2/CPU0
2001:3::19      48   0013.9400.0019 REACH BE3           0/2/CPU0
2001:3::1a      67   0013.9400.001a REACH BE3           0/2/CPU0
2001:3::1b      91   0013.9400.001b REACH BE3           0/2/CPU0
2001:3::1c      33   0013.9400.001c REACH BE3           0/2/CPU0
2001:3::1d      174  0013.9400.001d DELAY BE3           0/2/CPU0
2001:3::1e      144  0013.9400.001e REACH BE3           0/2/CPU0
2001:3::1f      121  0013.9400.001f REACH BE3           0/2/CPU0
2001:3::20      53   0013.9400.0020 REACH BE3           0/2/CPU0
```

This table describes significant fields shown in the display.

***Table 8: show ipv6 neighbors Command Field Descriptions***

| Field | Description |
|---|---|
| IPv6 Address | IPv6 address of neighbor or interface. |
| Age | Time (in minutes) since the address was confirmed to be reachable. A hyphen (-) indicates a static entry. |
| Link-layer Addr | MAC address. If the address is unknown, a hyphen (-) is displayed. |

| Field | Description |
|---|---|
| State | The state of the neighbor cache entry. These are the states for dynamic entries in the IPv6 neighbor discovery cache:<br>• INCMP (incomplete)—Address resolution is being performed on the entry. A neighbor solicitation message has been sent to the solicited-node multicast address of the target, but the corresponding neighbor advertisement message has not yet been received.<br>• reach (reachable)—Positive confirmation was received within the last ReachableTime milliseconds that the forward path to the neighbor was functioning properly. While in reach state, the device takes no special action as packets are sent.<br>• stale—More than ReachableTime milliseconds have elapsed since the last positive confirmation was received that the forward path was functioning properly. While in stale state, the device takes no action until a packet is sent.<br>• delay—More than ReachableTime milliseconds have elapsed since the last positive confirmation was received that the forward path was functioning properly. A packet was sent within the last DELAY_FIRST_PROBE_TIME seconds. If no reachability confirmation is received within DELAY_FIRST_PROBE_TIME seconds of entering the delay state, send a neighbor solicitation message and change the state to probe.<br>• probe—A reachability confirmation is actively sought by resending neighbor solicitation messages every RetransTimer milliseconds until a reachability confirmation is received.<br>These are the possible states for static entries in the IPv6 neighbor discovery cache:<br>• reach (reachable)—The interface for this entry is up.<br>• INCMP (incomplete)—The interface for this entry is down.<br>**Note**<br>Reachability detection is not applied to static entries in the IPv6 neighbor discovery cache; therefore, the descriptions for the INCMP (incomplete) and reach (reachable) states are different for dynamic and static cache entries. |
| Interface | Interface from which the address is reachable. |

# show ipv6 neighbors summary

To display summary information for the neighbor entries, use the **show ipv6 neighbors summary** command in the XR EXEC mode.

**show ipv6 neighbors summary**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The default value is disabled.

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read |

## Examples

This is the sample output of the **show ipv6 neighbors summary** command that shows the summary information for the neighbor entries:

```
XR EXEC mode# show ipv6 neighbors summary

Mcast nbr entries:
    Subtotal: 0
Static nbr entries:
    Subtotal: 0
Dynamic nbr entries:
    Subtotal: 0

Total nbr entries: 0
```

# show ipv6 path-mtu

To display path maximum transmission unit (MTU) details of IPv6 packets, use the **show ipv6 path-mtu** command in the XR Config mode.

**show ipv6 path-mtu** [ **vrf** { *vrf-name* | **all**} [ **location** *node-id* ] ] [ **location** *node-id* ]

## Syntax Description

| | |
|---|---|
| **location** *node-id* | (Optional) The designated node. The node-id argument is entered in the *rack/slot/module* notation. |

## Command Default

None.

## Command Modes

XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

If the location option is specified, only the details of the node specified in the **location** *node-id* keyword and argument are displayed. Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.

## Task ID

| Task ID | Operations |
|---|---|
| ipv6 | read, write |
| network | read, write |
| config-services | read, write |

## Examples

This example shows how to display path MTU details of IPv6 packets:

```
RP/0/RP0/CPU0:router(config)#  show ipv6 pmtu

Destination         Ifhandle        Vrfid              Path Mtu       Time Left
bb::1               0x300           0x60000000         1300           00:01:27
cd::1               0x300           0x60000000         1300           00:01:42
```

# show ipv6 traffic

To display the IPv6 traffic statistics, use the **show traffic** command in the XR EXEC mode.

**show ipv6 traffic** [**brief**]

**Syntax Description**

| | |
|---|---|
| **brief** | (Optional) Displays only IPv6 and Internet Control Message Protocol version 6 (ICMPv6) traffic statistics. |

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

**Usage Guidelines**

The **show ipv6 traffic** command provides output similar to the **show ipv4 traffic** command, except that it is IPv6-specific.

**Task ID**

| Task ID | Operations |
|---|---|
| ipv6 | read |
| network | read |

**Examples**

This is the sample output of the **show ipv6 traffic** command:

```
RP/0/RP0/CPU0:router# show ipv6 traffic

 IPv6 statistics:
  Rcvd:  0 total, 0 local destination
         0 source-routed, 0 truncated
         0 format errors, 0 hop count exceeded
         0 bad header, 0 unknown option, 0 bad source
         0 unknown protocol
         0 fragments, 0 total reassembled
         0 reassembly timeouts, 0 reassembly failures
         0 reassembly max drop
         0  sanity address check drops
  Sent:  0 generated, 0 forwarded
         0 fragmented into 0 fragments, 0 failed
         0 no route, 0 too big
  Mcast: 0 received, 0 sent

ICMP statistics:
  Rcvd: 0 input, 0 checksum errors, 0 too short
        0 unknown error type
        unreach: 0 routing, 0 admin, 0 neighbor,
                 0 address, 0 port, 0 unknown
```

```
        parameter: 0 error, 0 header, 0 option,
                   0 unknown
        0 hopcount expired, 0 reassembly timeout,
        0 unknown timeout, 0 too big,
        0 echo request, 0 echo reply
  Sent: 0 output, 0 rate-limited
        unreach: 0 routing, 0 admin, 0 neighbor,
                 0 address, 0 port, 0 unknown
        parameter: 0 error, 0 header, 0 option
                 0 unknown
        0 hopcount expired, 0 reassembly timeout,
        0 unknown timeout, 0 too big,
        0 echo request, 0 echo reply

Neighbor Discovery ICMP statistics:
  Rcvd: 0 router solicit, 0 router advert, 0 redirect
        0 neighbor solicit, 0 neighbor advert
  Sent: 0 router solicit, 0 router advert, 0 redirect
        0 neighbor solicit, 0 neighbor advert

UDP statistics:
        0 packets input, 0 checksum errors
        0 length errors, 0 no port, 0 dropped
        0 packets output

TCP statistics:s
        0 packets input, 0 checksum errors, 0 dropped
        0 packets output, 0 retransmitted
```

This table describes the significant fields shown in the display.

***Table 9: show ipv6 traffic Command Field Descriptions***

| Field | Description |
|---|---|
| Rcvd: | Statistics in this section refer to packets received by the router. |
| total | Total number of packets received by the software. |
| local destination | Locally destined packets received by the software. |
| source-routed | Packets seen by the software with RH. |
| truncated | Truncated packets seen by the software. |
| bad header | An error was found in generic HBH, RH, DH, or HA. Software only. |
| unknown option | Unknown option type in IPv6 header. |
| unknown protocol | Protocol specified in the IP header of the received packet is unreachable. |
| Sent: | Statistics in this section refer to packets sent by the router. |
| forwarded | Packets forwarded by the software. If the packet cannot be forwarded in the first lookup (for example, the packet needs option processing), then the packet is not included in this count, even if it ends up being forwarded by the software. |
| Mcast: | Multicast packets. |

| Field | Description |
|---|---|
| ICMP statistics: | Internet Control Message Protocol statistics. |