# Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6), which includes addressing, Neighbor Discovery Protocol (ND), and Internet Control Message Protocol version 6 (ICMPv6), on the Cisco NX-OS device.

This chapter includes the following sections:

# About IPv6

IPv6, which is designed to replace IPv4, increases the number of network address bits from 32 bits (in IPv4) to 128 bits. IPv6 is based on IPv4 but it includes a much larger address space and other improvements such as a simplified main header and extension headers.

The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. The flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT), which translates private (not globally unique) addresses into a limited number of public addresses. IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 functionality, such as prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities, enable more efficient routing. IPv6 supports Open Shortest Path First (OSPF) for IPv6 and multiprotocol Border Gateway Protocol (BGP).

## IPv6 Address Formats

An IPv6 address has 128 bits or 16 bytes. The address is divided into eight, 16-bit hexadecimal blocks separated by colons (:) in the format: x:x:x:x:x:x:x:x. Two examples of IPv6 addresses are as follows:

```
2001:0DB8:7654:3210:FEDC:BA98:7654:32102001:0DB8:0:0:8:800:200C:417A
```

IPv6 addresses contain consecutive zeros within the address. You can use two colons (::) at the beginning, middle, or end of an IPv6 address to replace the consecutive zeros. The following table shows a list of compressed IPv6 address formats.

**Note** You can use two colons (::) only once in an IPv6 address to replace the longest string of consecutive zeros within the address.

You can use a double colon as part of the IPv6 address when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interface but only one link-local address.

The hexadecimal letters in IPv6 addresses are not case sensitive.

*Table 1: Compressed IPv6 Address Formats*

| IPv6 Address Type | Preferred Format | Compressed Format |
|---|---|---|
| Unicast | 2001:0:0:0:0DB8:800:200C:417A | 2001::0DB8:800:200C:417A |
| Multicast | FF01:0:0:0:0:0:0:101 | FF01::101 |
| Loopback | 0:0:0:0:0:0:0:1 | ::1 |
| Unspecified | 0:0:0:0:0:0:0:0 | :: |

A node may use the loopback address listed in the table **Compressed IPv6 Address Formats** to send an IPv6 packet to itself. The loopback address in IPv6 is the same as the loopback address in IPv4. For more information, see Overview.

**Note** You cannot assign the IPv6 loopback address to a physical interface. A packet that contains the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

**Note** You cannot assign an IPv6 unspecified address to an interface. You should not use the unspecified IPv6 addresses as destination addresses in IPv6 packets or the IPv6 routing header.

The IPv6-prefix is in the form documented in RFC 2373 where the IPv6 address is specified in hexadecimal using 16-bit values between colons. The prefix length is 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). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

# IPv6 Unicast Addresses

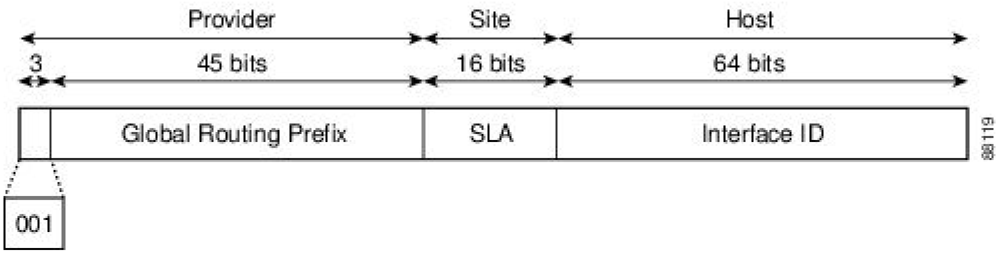
An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address.

## Aggregatable Global Addresses

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses are used on links that are aggregated upward through organizations and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). The following figure shows the structure of an aggregatable global address.

*Figure 1: Aggregatable Global Address Format*



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields called Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLS and NLA fields from the RFCs because these fields are policy based. Some existing IPv6 networks deployed before the change might still use networks that are on the older architecture.

A subnet ID, which is a 16-bit subnet field, can be used by individual organizations to create a local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID identifies interfaces on a link. The interface ID is unique to the link. In many cases, an interface ID is the same as or based on the link-layer address of an interface. Interface IDs used in aggregatable global unicast and other IPv6 address types have 64 bits and are in the modified EUI-64 format.

Interface IDs are in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and Fiber Distributed Data interfaces), the first three octets (24 bits) are the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are the last three octets of the MAC address. The Universal/Local (U/L) bit, which is the seventh bit of the first octet, has a value of 0 or 1. Zero indicates a locally administered identifier; 1 indicates a globally unique IPv6 interface identifier.

- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel interface types—except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is similar to the interface ID for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used as the identifier (because the interface does not have a MAC address).

- For tunnel interface types that are used with IPv6 overlay tunnels, the interface ID is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.

**Note** For interfaces that use the Point-to-Point Protocol (PPP), where the interfaces at both ends of the connection might have the same MAC address, the interface identifiers at both ends of the connection are negotiated (picked randomly and, if necessary, reconstructed) until both identifiers are unique. The first MAC address in the router is used as the identifier for interfaces using PPP.

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

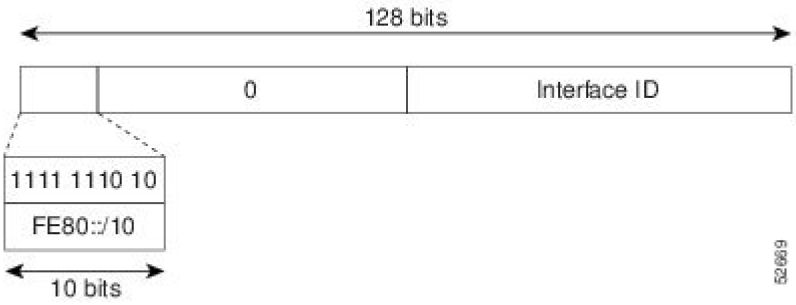
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

## Link-Local Addresses

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the Neighbor Discovery Protocol (NDP). Nodes on a local link can use link-local addresses to communicate; the nodes do not need globally unique addresses to communicate. The following figure shows the structure of a link-local address.

IPv6 routers cannot forward packets that have link-local source or destination addresses to other links.

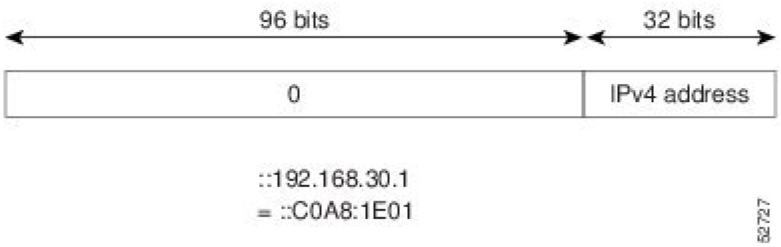
*Figure 2: Link-Local Address Format*



## IPv4-Compatible IPv6 Addresses

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32 bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels. The following figure shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

*Figure 3: IPv4-Compatible IPv6 Address Format*



## Unique Local Addresses

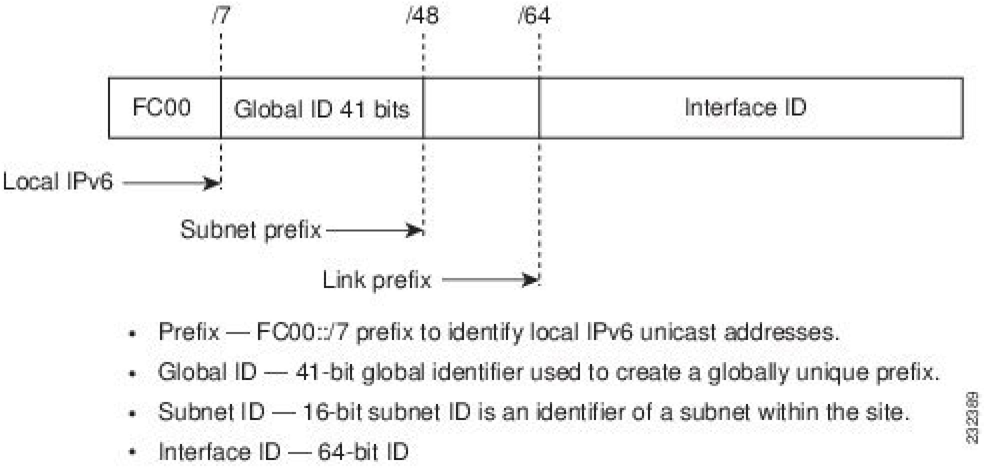
A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications may treat unique local addresses like global scoped addresses.

A unique local address has the following characteristics:

- It has a globally unique prefix (it has a high probability of uniqueness).
- It has a well-known prefix to allow for easy filtering at site boundaries.
- It allows sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- It is ISP-independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- If it is accidentally leaked outside of a site through routing or the Domain Name Server (DNS), there is no conflict with any other addresses.

The following figure shows the structure of a unique local address.

*Figure 4: Unique Local Address Structure*



- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit ID

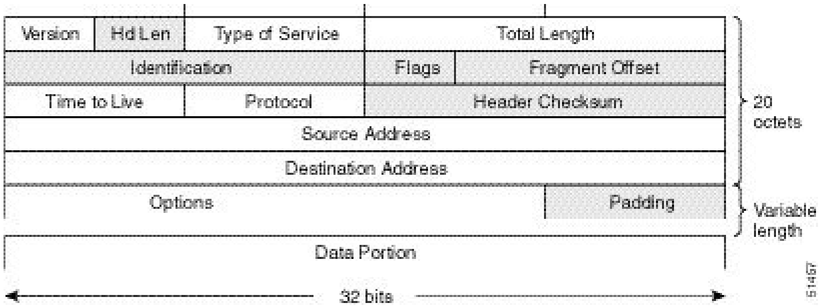
## Site-Local Address

Because RFC 3879 deprecates the use of site-local addresses, you should follow the recommendations of unique local addressing (ULA) in RFC 4193 when you configure private IPv6 addresses.

# IPv4 Packet Header

The base IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see the following figure). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header are not included in the IPv6 packet header.

*Figure 5: IPv4 Packet Header Format*



# Simplified IPv6 Packet Header

The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see the following figure). Fragmentation is handled by the source of a packet and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet and the base IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

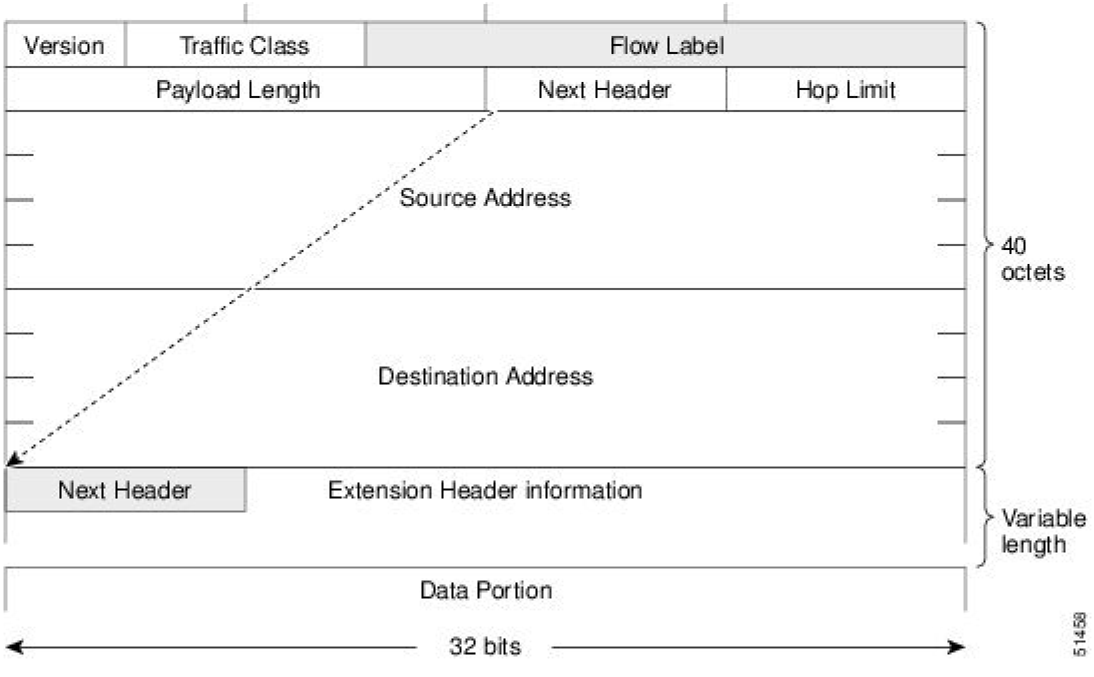
The following table lists the fields in the base IPv6 packet header.

*Table 2: Table 3-2 Base IPv6 Packet Header Fields*

| Field | Description |
|---|---|
| Version | Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4. |
| Traffic Class | Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services. |
| Flow Label | New field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer. |
| Payload Length | Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet. |

| Field | Description |
|---|---|
| Next Header | Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information that follows the base IPv6 header. The type of information that follows the base IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in the following figure. |
| Hop Limit | Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources. |
| Source Address | Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4. |
| Destination Address | Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4. |

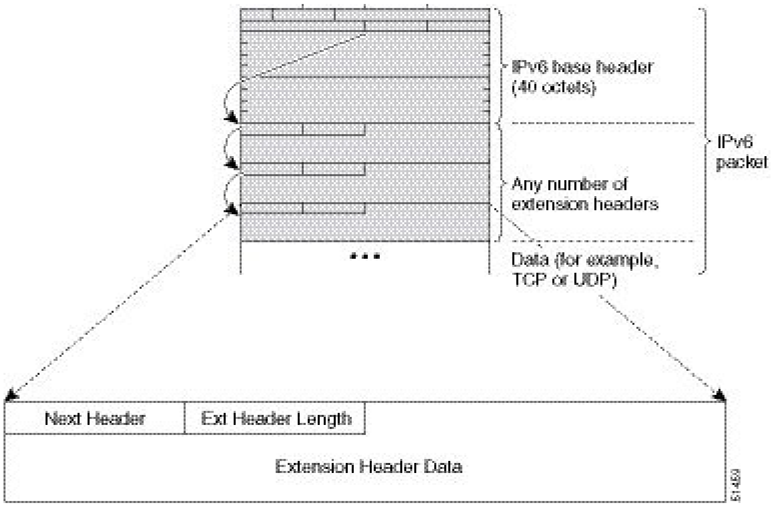
*Figure 6: IPv6 Packet Header Format*



### IPv6 Extension Headers

Optional extension headers and the data portion of the packet are after the eight fields of the base IPv6 packet header. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. The following figure shows the IPv6 extension header format.

*Figure 7: IPv6 Extension Header Format*



The following table lists the extension header types and their Next Header field values.

*Table 3: IPv6 Extension Header Types*

| Header Type | Next Header Value | Description |
|---|---|---|
| Hop-by-hop options header | 0 | Header that is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the base IPv6 packet header. |
| Destination options header | 60 | Header that can follow any hop-by-hop options header. The header is processed at the final destination and at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header. The destination options header is processed only at the final destination. |
| Routing header | 43 | Header that is used for source routing. |
| Fragment header | 44 | Header that is used when a source fragments a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet. |
| Upper-layer headers | 6 (TCP)<br>17 (UDP) | Headers that are used inside a packet to transport the data. The two main transport protocols are TCP and UDP. |

**Note** Some switch models support only a subset of IPv6 extension header types. The following list shows the extension header types that are supported by Cisco Nexus 3600 Platform Switches (N3K-C36180YC-R and N3K-C3636C-R) and by Cisco Nexus 9504 and 9508 modular chassis with these line cards: N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R.

Supported: Destination options (60), Routing (43), Fragment (44), Mobility (135), Host Identity Protocol (HIP) (139), Shim6 (140).

Not supported: Hop-by-hop options (0), Encapsulation Security Payload (50), Authentication Header (51), and experimental (253 and 254).

Beginning with Cisco NX-OS Release 9.3(7), if you configure an IPv6 ACL on the devices listed here, you must include a new rule for the disposition of IPv6 packets that include extension headers. For the necessary configuration procedure, see "Configuring an ACL for IPv6 Extension Headers" in NX-OS Release 9.3(x) or later of the *Cisco Nexus 3600 NX-OS Security Configuration Guide*.

# DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses (see the following table).

**Note** IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

*Table 4: IPv6 DNS Record Types*

| Record Type | Description | Format |
|---|---|---|
| AAAA | Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.) | www.abc.test AAAA 3FFE:YYYY:C18:1::2 |
| PTR | Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.) | 2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test |

# Path MTU Discovery for IPv6

As in IPv4, you can use path MTU discovery in IPv6 to allow a host to dynamically discover and adjust to differences in the MTU size of every link along a data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently. Once the path MTU is reduced by the arrival of an ICMP Too Big message, Cisco NX-OS retains the lower value. The connection does not increase the segment size to gauge the throughput.

**Note** In IPv6, the minimum link MTU is 1280 octets. We recommend that you use an MTU value of 1500 octets for IPv6 links.

## CDP IPv6 Address Support

You can use the Cisco Discovery Protocol (CDP) IPv6 address support for the neighbor information feature to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

## ICMP for IPv6

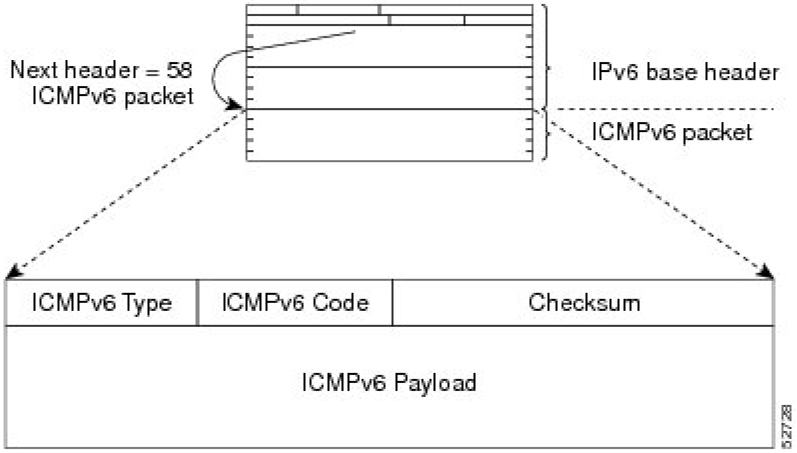
You can use ICMP in IPv6 to provide information about the health of the network. ICMPv6, the version that works with IPv6, reports errors if packets cannot be processed correctly and sends informational messages about the status of the network. For example, if a router cannot forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures that a packet is sent using the largest possible size that is supported on a specific route.

A value of 58 in the Next Header field of the base IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet.Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is computed by the sender and checked by the receiver from the fields in the IPv6 ICMP packet and the IPv6 pseudo header.

**Note** The IPv6 header does not have a checksum. But a checksum on the transport layer can determine if packets have not been delivered correctly. All checksum calculations that include the IP address in the calculation must be modified for IPv6 to accommodate the new 128-bit address. A checksum is generated using a pseudo header.

The ICMPv6 Payload field contains error or diagnostic information that relates to IP packet processing. The following figure shows the IPv6 ICMP packet header format.

*Figure 8: IPv6 ICMP Packet Header Format*



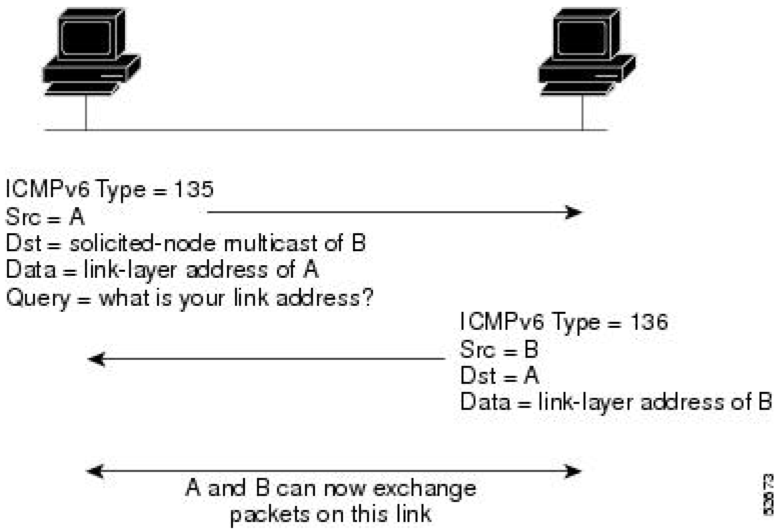
# IPv6 Neighbor Discovery

You can use the IPv6 Neighbor Discovery Protocol (NDP) to determine whether a neighboring router is reachable. IPv6 nodes use neighbor discovery to determine the addresses of nodes on the same network (local link), to find neighboring routers that can forward their packets, to verify whether neighboring routers are reachable or not, and to detect changes to link-layer addresses. NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

# IPv6 Neighbor Solicitation Message

A node sends a neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, on the local link when it wants to determine the link-layer address of another node on the same local link (see the following figure). The source address is the IPv6 address of the node that sends the neighbor solicitation message. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

*Figure 9: IPv6 Neighbor Discovery—Neighbor Solicitation Message*


ICMPv6 Type = 135
Src = A
Dst = solicited-node multicast of B
Data = link-layer address of A
Query = what is your link address?
ICMPv6 Type = 136
Src = B
Dst = A
Data = link-layer address of B
A and B can now exchange packets on this link

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address is the IPv6 address of the node (the IPv6 address of the node interface that sends the neighbor advertisement message). The destination address is the IPv6 address of the node that sends the neighbor solicitation message. The data portion includes the link-layer address of the node that sends the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages can verify the reachability of a neighbor after a node identifies the link-layer address of a neighbor. When a node wants to verify the reachability of a neighbor, it uses the destination address in a neighbor solicitation message as the unicast address of the neighbor.

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

Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination). If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

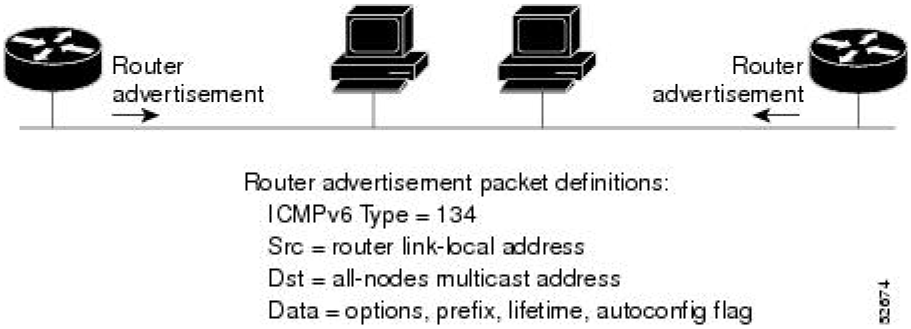
**Note** A neighbor advertisement message that has the solicited flag set to a value of 0 is not considered as a positive acknowledgment that the forward path is still working.

# Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out to each configured interface of an IPv6 router.

The RA messages are sent to the all-nodes multicast address (see the following figure).

***Figure 10: IPv6 Neighbor Discovery—RA Message***


Router advertisement
Router advertisement
Router advertisement packet definitions:
ICMPv6 Type = 134
Src = router link-local address
Dst = all-nodes multicast address
Data = options, prefix, lifetime, autoconfig flag
52674

RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Life-time information for each prefix included in the advertisement
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time in seconds that the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU that a host should use in packets that it originates

RAs are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message. The source address is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface that sends the router solicitation message is used as the source address in the message. The destination address is the all-routers multicast address with a scope of the link. When an RA is sent in response to a router solicitation, the destination address in the RA message is the unicast address of the source of the router solicitation message.

You can configure the following RA message parameters:

- The time interval between periodic RA messages

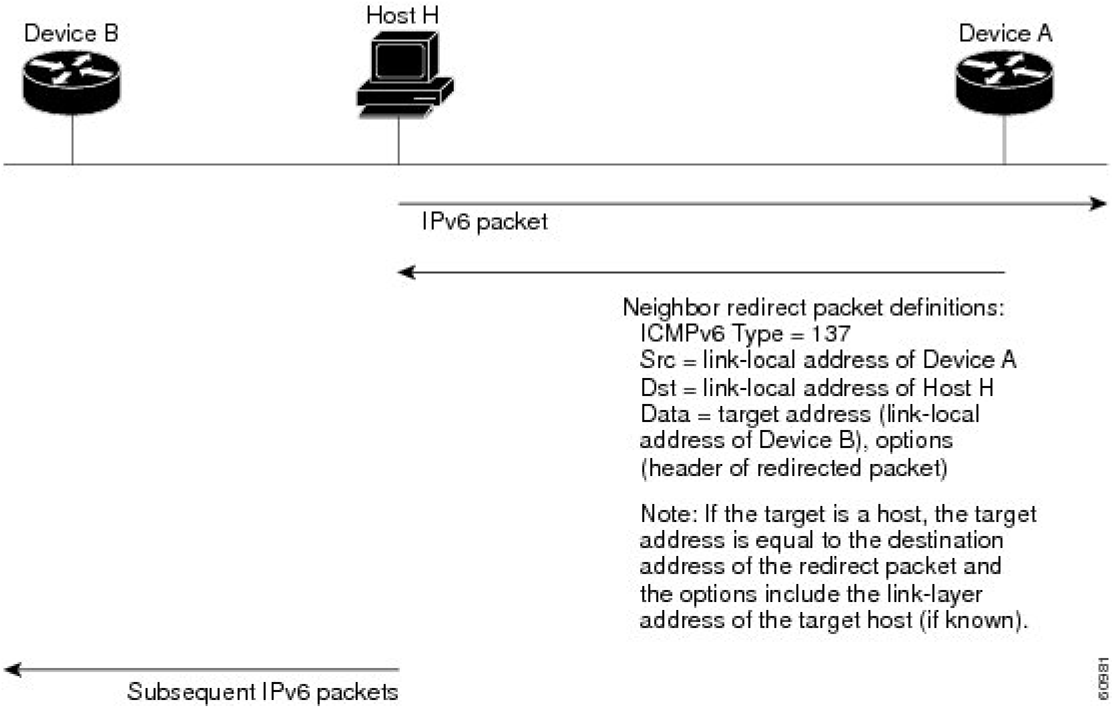
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time that a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on Ethernet interfaces. For other interface types, you must enter the **no ipv6 nd suppress-ra** command to send RA messages. You can disable the RA message feature on individual interfaces by entering the **ipv6 nd suppress-ra** command.

# IPv6 Neighbor Redirect Message

Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see the following figure). A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message.

***Figure 11: IPv6 Neighbor Discovery—Neighbor Redirect Message***


Device B
Host H
Device A
IPv6 packet
Neighbor redirect packet definitions:
ICMPv6 Type = 137
Src = link-local address of Device A
Dst = link-local address of Host H
Data = target address (link-local address of Device B), options (header of redirected packet)
Note: If the target is a host, the target address is equal to the destination address of the redirect packet and the options include the link-layer address of the target host (if known).
Subsequent IPv6 packets
60981

**Note** A router must be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor router by its link-local address. For static routing, you should specify the address of the next-hop router using the link-local address of the router. For dynamic routing, you must configure all IPv6 routing protocols to exchange the link-local addresses of neighboring routers.

After forwarding a packet, a router sends a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the router.
- The packet is about to be sent out the interface on which it was received.
- The router determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link or a link-local address.

## Virtualization Support

IPv6 supports virtual routing and forwarding (VRF) instances.

## IPv6 Routes with ECMP

If all next-hops for a route are glean, drop, or punt, all next-hops are programmed as-is in the Multipath hardware table.

If some next-hops for a route are glean, drop, or punt, and the remaining next-hops are not, then only non glean, drop, or punt next-hops are programmed in the Multipath hardware table.

When a specific next-hop for ECMP route is resolved (ARP/IPV6 ND resolved), then the Multipath hardware table is updated accordingly.

# Prerequisites for IPv6

IPv6 has the following prerequisites:

- You must be familiar with IPv6 basics such as IPv6 addressing, IPv6 header information, ICMPv6, and the IPv6 Neighbor Discovery (ND) Protocol.
- Ensure that you follow the memory/processing guidelines when you make a device a dual-stack device (IPv4/IPv6).

# Guidelines and Limitations for IPv6

IPv6 has the following configuration guidelines and limitations:

- IPv6 packets are transparent to Layer 2 LAN switches because the switches do not examine Layer 3 packet information before forwarding IPv6 frames. IPv6 hosts can be directly attached to Layer 2 LAN switches.
- You can configure multiple IPv6 global addresses within the same prefix on an interface. However, multiple IPv6 link-local addresses on an interface are not supported.

- Because RFC 3879 deprecates the use of site-local addresses, you should configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.
- For Cisco Nexus 3600-R platform switches, internet-peering mode is only intended to be used with the prefix pattern as distributed in the global internet routing table. In this mode, other prefix distributions or patterns can operate, but not predictably. As a result, maximum achievable LPM/LEM scale is reliable only when the prefix patterns are actual internet prefix patterns. In internet-peering mode, if route prefix patterns other than those in the global internet routing table are used, the switch might not successfully achieve documented scalability numbers.

# Default Settings

The following table lists the default settings for IPv6 parameters.

*Table 5: Default IPv6 Parameters*

| Parameters | Default |
|---|---|
| ND reachable time | 0 milliseconds |
| Neighbor solicitation retransmit interval | 1000 milliseconds |

# Configuring IPv6

## Configuring IPv6 Addressing

You must configure an IPv6 address on an interface so that the interface can forward IPv6 traffic. When you configure a global IPv6 address on an interface, it automatically configures a link-local address and activates IPv6 for that interface.

### SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet number**
3. 
4. (Optional) **show ipv6 interface**
5. (Optional) **copy running-config startup-config**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 2</b></td><td><b>interface ethernet number</b><br><b>Example:</b><br><code>switch(config)# interface ethernet 2/3</code><br><code>switch(config-if)#</code></td><td>Enters interface configuration mode.</td></tr>
<tr><td><b>Step 3</b></td><td>
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Command</td><td>Purpose</td></tr>
<tr><td><b>ipv6 address</b> { <i>addr</i> [<b>eui64</b> ] [ <b>route-preference</b> <i>preference</i> ] [ <b>secondary</b> ] <b>tag</b> <i>tag-id</i> ]</td><td>Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.<br>Entering the <b>ipv6 address</b> command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID.</td></tr>
<tr><td><b>ipv6 address ipv6-address use-link-local-only</b></td><td>Entering the <b>ipv6 address use-link-local-only</b> command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.<br>This command enables IPv6 processing on an interface without configuring an IPv6 address.</td></tr>
</table>
<b>Example:</b><br><code>switch(config-if)# ipv6 address 2001:0DB8::1/10</code><br>or<br><code>switch(config-if)# ipv6 address use-link-local-only</code></td><td></td></tr>
<tr><td><b>Step 4</b></td><td>(Optional) <b>show ipv6 interface</b><br><b>Example:</b><br><code>switch(config-if)# show ipv6 interface</code></td><td>Displays interfaces configured for IPv6.</td></tr>
<tr><td><b>Step 5</b></td><td>(Optional) <b>copy running-config startup-config</b><br><b>Example:</b><br><code>switch(config-if)# copy running-config</code><br><code>startup-config</code></td><td>Saves this configuration change.</td></tr>
</table>

### Example

This example shows how to configure an IPv6 address:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 address ?
A:B::C:D/LEN IPv6 prefix format: xxxx:xxxx/ml, xxxx:xxxx::/ml,
xxxx::xx/128
use-link-local-only Enable IPv6 on interface using only a single link-local
address
switch(config-if)# ipv6 address 2001:db8::/64 eui64
```

This example shows how to display an IPv6 interface:

```
switch(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod: 36
IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
IPv6 subnet: 0dc3:0dc3:0000:0000:0000:0000:0000:0000/64
IPv6 link-local address: fe80::0218:baff:fed8:239d (default)
IPv6 multicast routing: disabled
IPv6 multicast groups locally joined:
ff02::0001:ffd8:239d ff02::0002 ff02::0001 ff02::0001:ffd8:239d
IPv6 multicast (S,G) entries joined: none
IPv6 MTU: 1500 (using link MTU)
IPv6 RP inbound packet-filtering policy: none
IPv6 RP outbound packet-filtering policy: none
IPv6 inbound packet-filtering policy: none
IPv6 outbound packet-filtering policy: none
IPv6 interface statistics last reset: never
IPv6 interface RP-traffic statistics: (forwarded/originated/consumed)
Unicast packets: 0/0/0
Unicast bytes: 0/0/0
Multicast packets: 0/0/0
Multicast bytes: 0/0/0
```

## Configuring LPM Internet-Peering Routing Mode

Beginning with Cisco NX-OS Release 9.3(1), you can configure LPM Internet-peering routing mode in order to support IPv4 and IPv6 LPM Internet route entries. This mode supports dynamic Trie (tree bit lookup) for IPv4 prefixes (with a prefix length up to /32) and IPv6 prefixes (with a prefix length up to /83). The Cisco Nexus 3600-R platform switches support this routing mode.

**Note** This configuration impacts both the IPv4 and IPv6 address families.

**Note** For LPM Internet-peering routing mode scale numbers, see the Cisco Nexus 3600 Series NX-OS Verified Scalability Guide. Cisco Nexus 3600-R platform switches in LPM Internet-peering mode scale out prectably only if they use internet-peering prefixes. If a Cisco Nexus 3600-R platform switch uses other prefix patterns, it might not achieve documented scalability numbers.

### SUMMARY STEPS

1. **configure terminal**
2. [**no**] **system routing template-internet-peering**
3. (Optional) **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | [**no**] **system routing template-internet-peering**<br>**Example:**<br>`switch(config)# system routing`<br>`template-internet-peering` | Puts the device in LPM Internet-peering routing mode to support IPv4 and IPv6 LPM Internet route entries. |
| **Step 3** | (Optional) **show system routing mode**<br>**Example:**<br>`switch(config)# show system routing mode`<br>`Configured System Routing Mode: Internet Peering`<br>`Applied System Routing Mode: Internet Peering` | Displays the LPM routing mode. |
| **Step 4** | **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Saves this configuration change. |
| **Step 5** | **reload**<br>**Example:**<br>`switch(config)# reload` | Reboots the entire device. |

## Additional Configuration for LPM Internet-Peering Routing Mode

When you deploy a Cisco Nexus switch in LPM Internet-peering routing mode in a large-scale routing environment or for routes with an increased number of next hops, you need to increase the memory limits for IPv4 under the VDC resource template.

### SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show routing ipv4 memory estimate routes routes next-hops hops**
3. **vdc switch id id**
4. **limit-resource u4route-mem minimum min-limit maximum max-limit**
5. **exit**

6. **copy running-config startup-config**
7. **reload**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | (Optional) **show routing ipv4 memory estimate routes routes next-hops hops**<br>**Example:**<br>`switch(config)# show routing ipv4 memory estimate`<br>` routes 262144 next-hops 32`<br>`Shared memory estimates:`<br>`Current max 512 MB; 78438 routes with 64 nhs`<br>`in-use 2 MB; 2642 routes with 1 nhs (average)`<br>`Configured max 512 MB; 78438 routes with 64 nhs`<br>`Estimate memory with fixed overhead: 1007 MB;`<br>`262144 routes with 32 nhs`<br>`Estimate with variable overhead included:`<br>`- With MVPN enabled VRF: 1136 MB`<br>`- With OSPF route (PE-CE protocol): 1375 MB`<br>`- With EIGRP route (PE-CE protocol): 1651 M` | Displays shared memory estimates to help you determine the memory requirements for routes. |
| **Step 3** | **vdc switch id id**<br>**Example:**<br>`switch(config)# vdc switch id 1`<br>`switch(config-vdc)#` | Specifies the VDC switch ID. |
| **Step 4** | **limit-resource u4route-mem minimum min-limit maximum max-limit**<br>**Example:**<br>`switch(config-vdc)# limit-resource u4route-mem`<br>`minimum 1024 maximum 1024` | Configures the limits for IPv4 memory in megabytes. |
| **Step 5** | **exit**<br>**Example:**<br>`switch(config-vdc)# exit`<br>`switch(config)#` | Exits the VDC configuration mode. |
| **Step 6** | **copy running-config startup-config**<br>**Example:**<br>`switch(config)# copy running-config startup-config` | Saves this configuration change. |
| **Step 7** | **reload**<br>**Example:**<br>`switch(config)# reload` | Reboots the entire device. |

# Configuring IPv6 Neighbor Discovery

You can configure IPv6 neighbor discovery on the router. NDP enables IPv6 nodes and routers to determine the link-layer address of a neighbor on the same link, find neighboring routers, and keep track of neighbors.

### Before you begin

You must first enable IPv6 on the interface.

## SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet number**
3. **ipv6 nd** [ **hop-limit** *hop-limit* | **managed-config-flag** | **mtu** *mtu* | **ns-interval** *interval* | **other-config-flag** | **prefix** | **ra-interval** *interval* | **ra-lifetime** *lifetime* | **reachable-time** *time* | **redirects** | **retrans-timer** *time* | **suppress-ra** ]
4. (Optional) **show ipv6 nd interface**
5. (Optional) **copy running-config startup-config**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | **interface ethernet number**<br>**Example:**<br>`switch(config)# interface ethernet 2/31`<br>`switch(config-if)#` | Enters interface configuration mode. |
| **Step 3** | **ipv6 nd** [ **hop-limit** *hop-limit* \| **managed-config-flag** \| **mtu** *mtu* \| **ns-interval** *interval* \| **other-config-flag** \| **prefix** \| **ra-interval** *interval* \| **ra-lifetime** *lifetime* \| **reachable-time** *time* \| **redirects** \| **retrans-timer** *time* \| **suppress-ra** ]<br>**Example:**<br>`switch(config-if)# ipv6 nd prefix` | Neighbor discovery is enabled automatically when you configure an IPv6 address. This command enables the following additional IPv6 neighbor discovery options on the interface:<br>• **hop-limit** *hop-limit* —Advertises the hop limit in IPv6 neighbor discovery packets. The range is from 0 to 255.<br>• **managed-config-flag** —Advertises in ICMPv6 router-advertisement messages to use stateful address auto configuration to obtain address information.<br>• **mtu** *mtu* —Advertises the maximum transmission unit (MTU) in ICMPv6 router-advertisement messages on this link. The range is from 1280 to 65535 bytes.<br>• **ns-interval** *interval* —Configures the retransmission interval between IPv6 neighbor solicitation messages. The range is from 1000 to 3600000 milliseconds. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **other-config-flag** —Indicates in ICMPv6 router-advertisement messages that hosts use stateful auto configuration to obtain nonaddress related information.<br>• **prefix** —Advertises the IPv6 prefix in the router-advertisement messages.<br>• **ra-interval** *interval* —Configures the interval between sending ICMPv6 router-advertisement messages. The range is from 4 to 1800 seconds.<br>• **ra-lifetime** *lifetime* —Advertises the lifetime of a default router in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds.<br>• **reachable-time** *time* —Advertises the time when a node considers a neighbor up after receiving a reachability confirmation in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds.<br>• **redirects** —Enables sending ICMPv6 redirect messages.<br>• **retrans-timer** *time* —Advertises the time between neighbor-solicitation messages in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds.<br>• **suppress-ra** —Disables sending ICMPv6 router-advertisement messages. |
| **Step 4** | (Optional) **show ipv6 nd interface**<br>**Example:**<br>`switch(config-if)# show ipv6 nd interface` | Displays interfaces configured for IPv6 neighbor discovery. |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-if)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure IPv6 neighbor discovery reachable time:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 nd reachable-time 10
```

This example shows how to display an IPv6 neighbor discovery interface:

```
switch(config-if)# show ipv6 nd interface ethernet 3/1
ICMPv6 ND Interfaces for VRF "default"
Ethernet3/1, Interface status: protocol-down/link-down/admin-down
IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
ICMPv6 active timers:
Last Neighbor-Solicitation sent: never
Last Neighbor-Advertisement sent: never
Last Router-Advertisement sent:never
Next Router-Advertisement sent in: 0.000000
Router-Advertisement parameters:
Periodic interval: 200 to 600 seconds
Send "Managed Address Configuration" flag: false
Send "Other Stateful Configuration" flag: false
Send "Current Hop Limit" field: 64
Send "MTU" option value: 1500
Send "Router Lifetime" field: 1800 secs
Send "Reachable Time" field: 10 ms
Send "Retrans Timer" field: 0 ms
Neighbor-Solicitation parameters:
NS retransmit interval: 1000 ms
ICMPv6 error message parameters:
Send redirects: false
Send unreachables: false
```

# Optional IPv6 Neighbor Discovery

You can use the following optional IPv6 Neighbor Discovery commands:

| Command | Purpose |
|---|---|
| **ipv6 nd hop-limit** | Configures the maximum number of hops used in router advertisements and all IPv6 packets that are originated by the router. |
| **ipv6 nd managed-config-flag** | Sets the managed address configuration flag in IPv6 router advertisements. |
| **ipv6 nd mtu** | Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface. |
| **ipv6 nd ns-interval** | Configures the interval between IPv6 neighbor solicitation retransmissions on an interface. |
| **ipv6 nd other-config-flag** | Configures the other stateful configuration flag in IPv6 router advertisements. |
| **ipv6 nd ra-interval** | Configures the interval between IPv6 router advertisement (RA) transmissions on an interface. |
| **ipv6 nd ra-lifetime** | Configures the router lifetime value in IPv6 router advertisements on an interface. |
| **ipv6 nd reachable-time** | Configures the amount of time that a remote IPv6 node is considered reachable after some reachability confirmation event has occurred. |
| **ipv6 nd redirects** | Enables ICMPv6 redirect messages to be sent. |
| **ipv6 nd retrans-timer** | Configures the advertised time between neighbor solicitation messages in router advertisements. |

| Command | Purpose |
|---|---|
| **ipv6 nd suppress-ra** | Suppresses IPv6 router advertisement transmissions on a LAN interface. |

## Configuring IPv6 Packet Verification

Cisco NX-OS supports an Intrusion Detection System (IDS) that checks for IPv6 packet verification. You can enable or disable these IDS checks.

**Note** Cisco Nexus 3600 platform switches do not filter out packets with a source IP address of 0.0.0.0.

To enable IDS checks, use the following commands in global configuration mode:

| Command | Purpose |
|---|---|
| **hardware ip verify address** { **destination zero** \| **identical** \| **reserved** \| **source multicast** } | Performs the following IDS checks on the IPv6 address:<br>• **destination zero** —Drops IPv6 packets if the destination IP address is ::.<br>• **identical** —Drops IPv6 packets if the source IPv6 address is identical to the destination IPv6 address.<br>• **reserved** —Drops IPv6 packets if the IPv6 address is ::1.<br>• **source multicast** —Drops IPv6 packets if the IPv6 source address is in the FF00::/8 range (multicast). |
| **hardware ipv6 verify length** { **consistent** \| **maximum** { max-frag \| **max-tcp** \| **udp** }} | Performs the following IDS checks on the IPv6 address:<br>• **consistent** —Drops IPv6 packets where the Ethernet frame size is greater than or equal to the IPv6 packet length plus the Ethernet header.<br>• **maximum max-frag** —Drops IPv6 packets if the formula (IPv6 Payload Length – IPv6 Extension Header Bytes) + (Fragment Offset * 8) is greater than 65536.<br>• **maximum max-tcp** —Drops IPv6 packets if the TCP length is greater than the IP payload length.<br>• **maximum udp** —Drops IPv6 packets if the IPv6 payload length is less than the UDP packet length. |
| **hardware ipv6 verify tcp tiny-frag** | Drops TCP packets if the IPv6 fragment offset is 1, or if the IPv6 fragment offset is 0 and the IP payload length is less than 16. |
| **hardware ipv6 verify version** | Drops IPv6 packets if the EtherType is not set to 6 (IPv6). |

Use the **show hardware forwarding ip verify** command to display the IPv6 packet verification configuration.

# Verifying the IPv6 Configuration

To display the IPv6 configuration, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show hardware forwarding ip verify** | Displays the IPv4 and IPv6 packet verification configuration. |
| **show ipv6 interface** | Displays IPv6-related interface information. |
| **show ipv6 adjacency** | Displays the adjacency table. |
| **show ipv6 icmp** | Displays ICMPv6 information. |
| show ipv6 nd | Displays IPv6 neighbor discovery interface information. |
| show ipv6 neighbor | Displays IPv6 neighbor entry. |

# Configuration Examples for IPv6

This example shows how to configure IPv6:

```
configure terminal
interface ethernet 3/1
ipv6 address 2001:db8::/64 eui64
ipv6 nd reachable-time 10
```