



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:

- [Information About IPv6, on page 1](#)
- [Prerequisites for IPv6, on page 15](#)
- [Guidelines and Limitations for IPv6, on page 16](#)
- [Default Settings, on page 16](#)
- [Configuring IPv6, on page 16](#)
- [Verifying the IPv6 Configuration, on page 28](#)
- [Configuration Examples for IPv6, on page 28](#)
- [Additional References, on page 29](#)

Information 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:BA98:7654:3210
```

```
2001:0DB8:0:0:8:800:200C: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:DB8:800:200C:417A	2001::0DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:101	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 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.

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

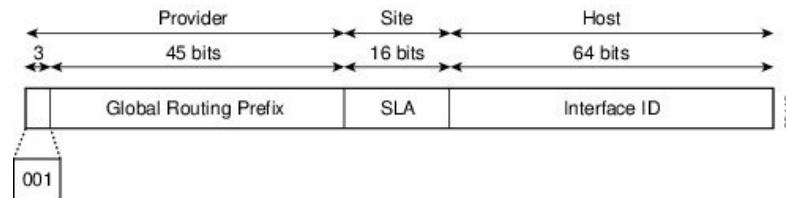
An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address.

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 TLA 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, and Frame Relay types), 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).



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:

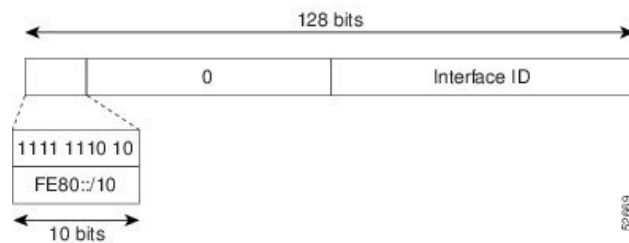
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, 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) and the stateless autoconfiguration process. 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.

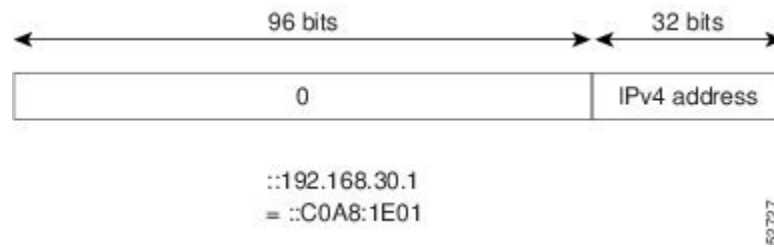
Figure 2: Link-Local Address Format



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: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. 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

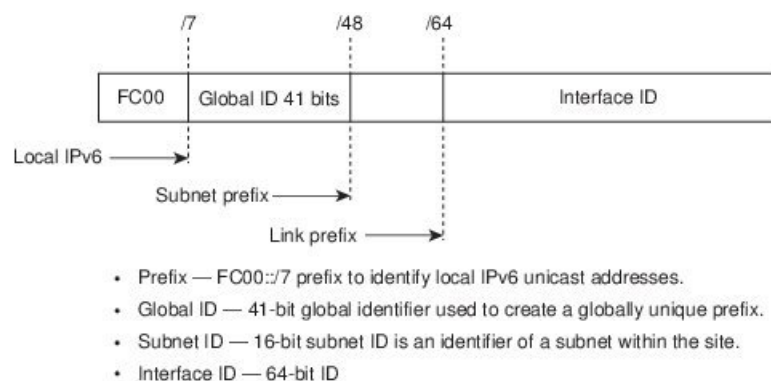
A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications might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



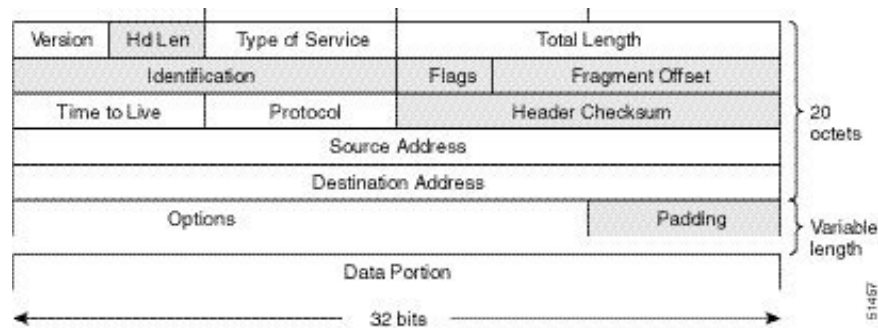
Site-Local Address

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 figure below). 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.

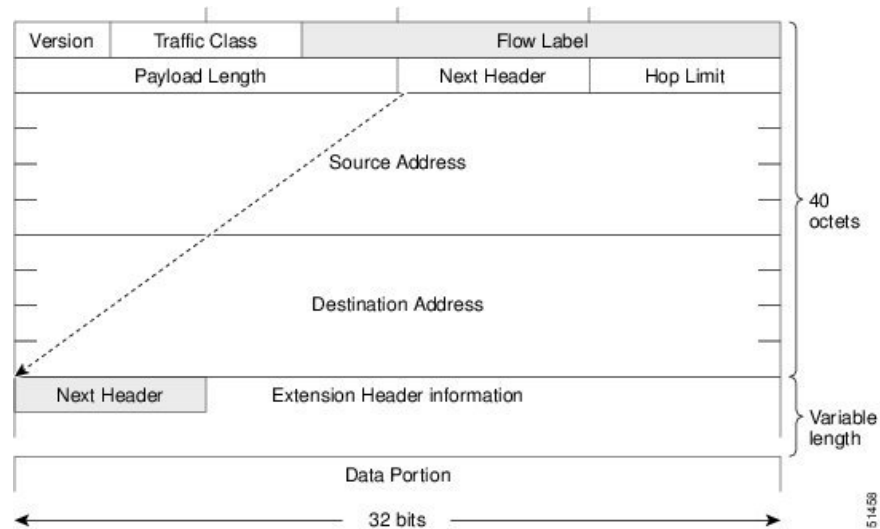
The following table lists the fields in the base IPv6 packet header.

Table 2: 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.

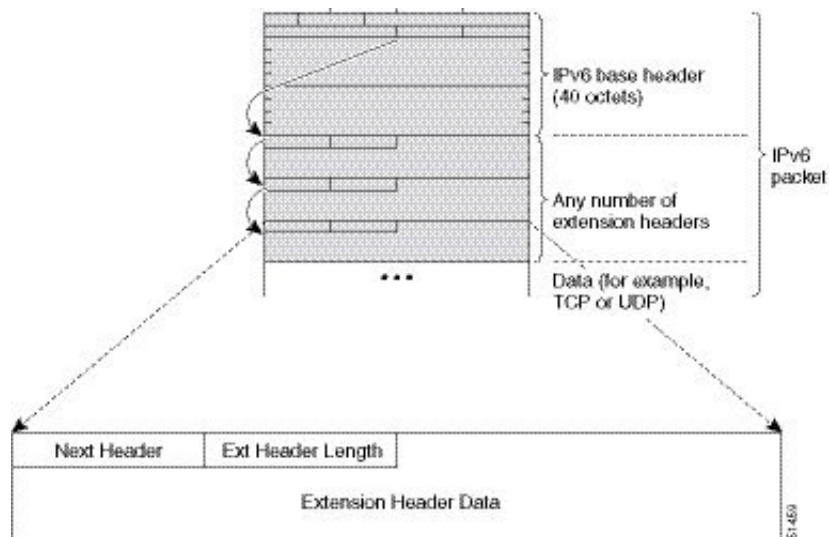
Field	Description
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information that follows the base IPv6 header. The type of information that follows the base IPv6 header can be a transport-layer packet (for example, a TCP or UDP packet) or an Extension Header, as shown in the figure below.
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



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 table below 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	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	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.
Routing	43	Header that is used for source routing.
Fragment	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.
Authentication	51	Header that is used to provide connectionless integrity and data origin authentication for packets.
Encapsulation Security Payload	50	All information following this header is encrypted.
Mobility	135	Header that is used in support of Mobile IPv6 service.
Host Identity Protocol	139	Header that is used for Host Identity Protocol version 2 (HIPv2), which provides secure methods for IP multihoming and mobile computing.



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.

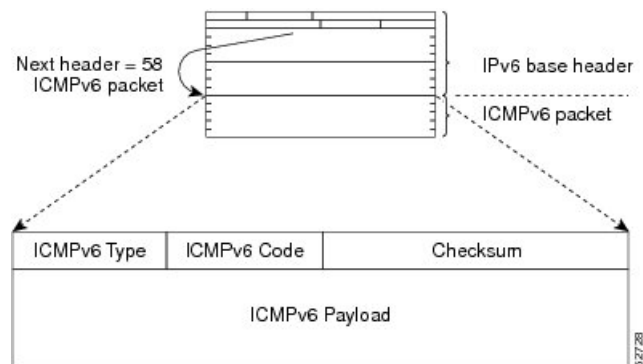
A value of 58 in the Next Header field of the base IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet 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



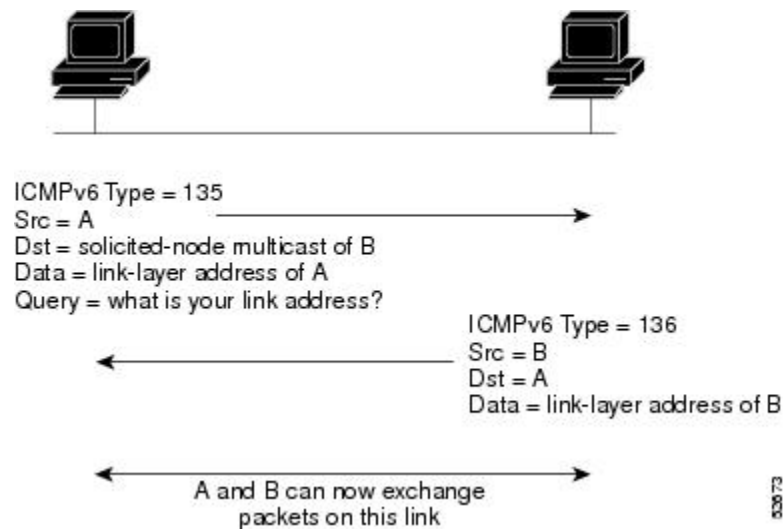
IPv6 Neighbor Discovery

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 figure below). 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



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.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). A node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

IPv6 Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

IPv6 Stateless Address Autoconfiguration (SLAAC) is performed only on a management interface. For example, when SLAAC is enabled on a management interface, it generates a Link Local Address (LLA) and performs a Duplicate Address Detection (DAD) on link local address. After the successful duplicate address detection process, the interface transmits ICMPv6 Router Solicitation (RS) packets. The upstream router that receives the RS packets responds back with an ICMPv6 Router Advertisement (RA). The RA packet will have a prefix TLV option that carries the subnet in which the downstream NX-OS Switch auto-generates the address, using the MAC information of the interface and the advertised prefix in RA packet. The Cisco NX-OS Switch auto-generates address in EUI-64 format and performs DAD on the new auto-generated addresses.

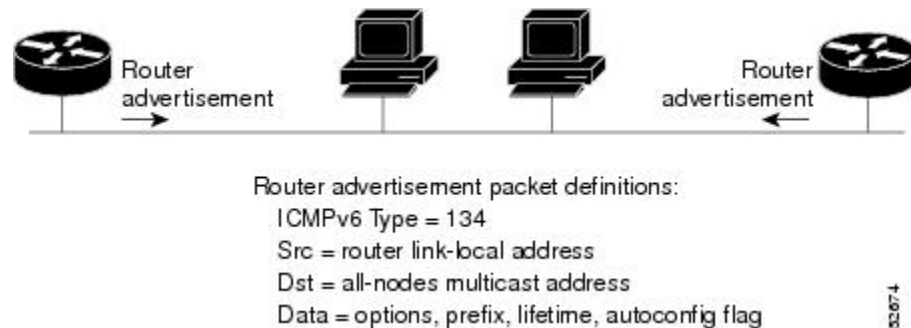
IPv6 addresses are assigned to an interface for a specific length of time. Each address has a lifetime that indicates how long the address is attached to an interface. The TLV prefix in the RA packet sent from the upstream router contain information about valid lifetime and preferred lifetime. The addresses that are assigned to an interface goes through two distinct phases. Initially, an address goes to a preferred state which means the address is not restricted for using in arbitrary communication. The address becomes deprecated state when the current interface binding becomes invalid. In a deprecated state, the use of the address is discouraged, not necessarily forbidden. Only applications that would have difficulty in switching to another address without a service disruption must use a deprecated address.

IPv6 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. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

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

Figure 10: IPv6 Neighbor Discovery-RA Message



The RA messages are sent to the all-nodes multicast address.

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
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time in seconds 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:

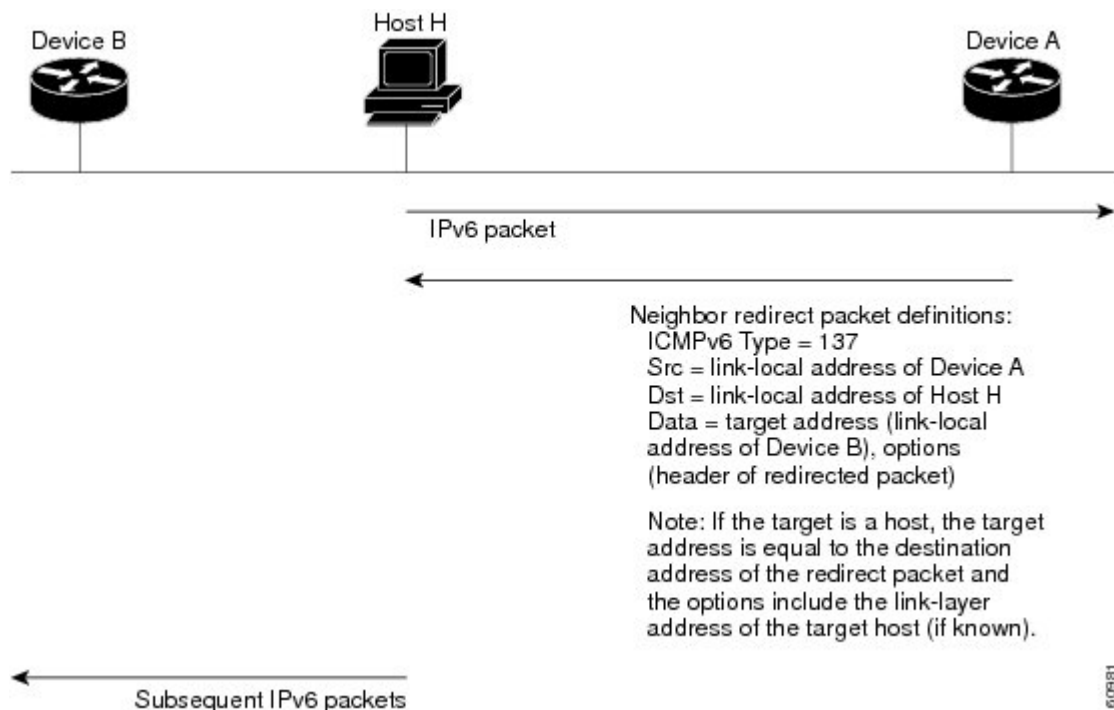
- The time interval between periodic RA messages
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (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. 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



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.

Interface peering Backup paths is not supported for IPv4 address-families learnt over IPv6 neighbors.

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.

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



Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

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.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 address {<i>address</i> [<i>eui64</i>] [<i>route-preference preference</i>] [<i>secondary</i>] [<i>tag tag-id</i>] or ipv6 address <i>ipv6-address</i> use-link-local-only Example: switch(config-if)# ipv6 address 2001:0DB8::1/10 or switch(config-if)# ipv6 address use-link-local-only	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. Entering the ipv6 address 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. Entering the ipv6 address use-link-local-only 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. This command enables IPv6 processing on an interface without configuring an IPv6 address.
Step 4	(Optional) show ipv6 interface Example: switch(config-if)# show ipv6 interface	Displays interfaces configured for IPv6.
Step 5	(Optional) copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	Saves this configuration change.

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: 2001:db8:0000:0000:0218:baff:fed8:239d
IPv6 subnet: 2001:db8::/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 the System Routing Mode

By default, the device programs routes in a hierarchical fashion (with fabric modules configured to be in mode 4 and line card modules configured to be in mode 3), which allows for longest prefix match (LPM) and host scale on the device.

You can modify the default LPM and host scale to program more hosts in the system, as might be required when the node is positioned as a Layer-2 to Layer-3 boundary node.

You must reload the switch by using the reload command to apply the configured system routing mode. You cannot use the fast-reload command to apply the configured system routing mode.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	[no] system routing max-mode host I3 Example: <pre>switch(config)# system routing max-mode 13 host</pre>	Enables algorithmic longest prefix match (ALPM) mode. When ALPM mode is enabled, you can carve the LPM table for IPv6 routes. The no form of this command disables ALPM.

	Command or Action	Purpose
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 4	reload Example: switch(config)# reload	Reboots the entire device.
Step 5	(Optional) show hardware profile status Example: switch(config)# show hardware profile status	Displays the updated number of LPM routes and the LPM mode.
Step 6	show running-config Example: switch(config)# show running-config	Displays the applied configuration.

Example

This example shows how to configure the system routing mode and enable ALPM:

```
switch# configure terminal
switch(config)# system routing max-mode 13
switch(config)# copy running-config startup-config
[#####] 100%
Copy complete, now saving to disk (please wait)...
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# show hardware profile status
Total LPM Entries = 65023.
Total Host Entries = 16384.
Reserved LPM Entries = 1024.
Max Host4/Host6 Limit Entries (shared)= 8192/4096*
Max Mcast Limit Entries = 4096.
Max Ucast IPv6 LPM Limit Entries = 20479.
Max Ucast IPv6 LPM_65_to_127 Limit Entries = 0.
Used LPM Entries (Total) = 55.
Used IPv4 LPM Entries = 27.
Used IPv6 LPM Entries = 28.
Used IPv6 LPM_65_to_127 Entries = 0.
Used Host Entries in LPM (Total) = 2.
Used Host4 Entries in LPM = 2.
Used Host6 Entries in LPM = 0.
Used Mcast Entries = 0.
Used Mcast OIFL Entries = 2.
Used Host Entries in Host (Total) = 152.
Used Host4 Entries in Host = 102.
Used Host6 Entries in Host = 50.
Max ECMP Table Entries = 256.
Used ECMP Table Entries = 0.
MFIB prefer-source-tree = Disabled/0/0.
```

*Unicast LPM Table is in ALPM mode shared b/n v4 & v6...
 *Unicast Host Table is in shared mode b/n v4 & v6...

Configuring ALPM

You can use the 64-bit algorithmic longest prefix match (ALPM) feature to manage IPv4 and IPv6 route table entries. In ALPM mode, the device can store significantly more route entries. Without IPv6 ALPM carving, you can program the route as follows:

URPF Status	IPv4 + IPv6-64 (/0-64)
Disabled	Upto 131071 IPv4 or upto 81919IPv6 /64
Enabled	Upto 65535 IPv4 or upto 20479 IPv6 /64



Note There is no support for IPv6-128(/65-/127) routes in this mode.

With IPv6 ALPM carving, you can program the route as follows:

Table 6:

ipv6_alpm_carve_value	IPv4 (non-URPF)/(URPF)	IPv6-64(/0-64) + IPv6-128(/65-127)
1028	98303/unsupported	5119/unsupported
2048	65535/32767	10239/2047
3072	32767/unsupported	12767/unsupported

All IPv6 routes including /65-127b routes share the same table. The IPv4 routes have dedicated memory.



Note In ALPM mode, only upto 40K IPv4 + 40K IPv6 with 16 way ECMP is supported with ECMP routes.

After you carve the LPM table, the carved space is used for all IPv6 routes with masks between /0 and /127. The remaining space is used only for IPv4 routes.

Without explicit IPv6 carving, the system supports IPv6 routes with masks between /0 and /64.



Note You must reload the switch by using the reload command to apply the configured carving values. You cannot use the fast-reload command to apply the configured carving values.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] hardware profile ipv6 alpm carve-value <i>ipv6_alpm_carve_value</i> Example: switch(config)# hardware profile ipv6 alpm 2048	Enables /128 IPv6 routes and carves the LPM table for all IPv6 entries.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 4	reload Example: switch(config)# reload	Reboots the entire device.
Step 5	(Optional) show hardware profile status Example: switch(config)# show hardware profile status	Displays the updated number of LPM routes and the LPM mode.
Step 6	show running-config Example: switch(config)# show running-config	Displays the applied configuration.

Example



Note The carve-value 1024 and 3072 are not supported when urpf is enabled.

This example shows how to configure LPM table carving:

```
switch# configure terminal
switch(config)# hardware profile ipv6 alpm carve-value 2048
Warning: Please copy running-config to startup-config and reload the switch to apply changes
switch(config)# copy running-config startup-config
[#####] 100%
Copy complete, now saving to disk (please wait)...
switch(config)# reload
```

```

WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# show hardware profile status
Max Ucast IPv4 LPM Limit Entries = 65535.
Total Host Entries = 16384.
Reserved LPM Entries = 1024.
Max Host4/Host6 Limit Entries (shared)= 8192/4096*
Max Mcast Limit Entries = 4096.
Max Ucast IPv6 LPM Limit Entries = 10239.
Used IPv4 LPM Entries = 65533.
Used IPv6 LPM Entries = 10239.
Used Host Entries in LPM (Total) = 2.
Used Host4 Entries in LPM = 2.
Used Host6 Entries in LPM = 0.
Used Mcast Entries = 0.
Used Mcast OIFL Entries = 2.
Used Host Entries in Host (Total) = 21.
Used Host4 Entries in Host = 19.
Used Host6 Entries in Host = 2.
Max ECMP Table Entries = 256.
Used ECMP Table Entries = 0.
MFIB prefer-source-tree = Disabled/0/0.

*Unicast LPM Table is in ALPM mode with ipv6 carving...
*IPv6 LPM and LPM 65 TO 127 tables are in shared mode...
*Unicast Host Table is in shared mode b/n v4 & v6...

```

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.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 nd [hop-limit <i>hop-limit</i> managed-config-flag mtu <i>mtu</i> ns-interval <i>interval</i> other-config-flag prefix ra-interval <i>interval</i> ra-lifetime <i>lifetime</i> 	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.

	Command or Action	Purpose
	<p>reachable-time <i>time</i> redirects retrans-timer <i>time</i> suppress-ra]</p> <p>Example:</p> <pre>switch(config-if)# ipv6 nd prefix</pre>	<ul style="list-style-type: none"> • hop-limit— Advertises the hop limit in IPv6 neighbor discovery packets. The range is from 0 to 255. • managed-config-flag— Advertises in ICMPv6 router-advertisement messages to use stateful address auto-configuration to obtain address information. • mtu— Advertises the maximum transmission unit (MTU) in ICMPv6 router-advertisement messages on this link. The range is from 1280 to 65535 bytes. • ns-interval— Configures the retransmission interval between IPv6 neighbor solicitation messages. The range is from 1000 to 3600000 milliseconds. • other-config-flag— Indicates in ICMPv6 router-advertisement messages that hosts use stateful auto configuration to obtain nonaddress related information. • prefix— Advertises the IPv6 prefix in the router-advertisement messages. • ra-interval— Configures the interval between sending ICMPv6 router-advertisement messages. The range is from 4 to 1800 seconds. • ra-lifetime— Advertises the lifetime of a default router in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • reachable-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. • redirects— Enables sending ICMPv6 redirect messages. <p>Note When disabling IPv6 redirects, IPv4 redirects should also be disabled as some IPv6 packets may still be leaked to the CPU.</p> <ul style="list-style-type: none"> • retrans-timer— time-Advertises the time between neighbor-solicitation messages in

	Command or Action	Purpose
		ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. <ul style="list-style-type: none"> • suppress-ra— Disables sending ICMPv6 router-advertisement messages.
Step 4	(Optional) show ip nd interface Example: switch(config-if)# show ip interface	Displays interfaces configured for IPv6 neighbor discovery.
Step 5	(Optional) copy running-config startup-config Example: 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 interface:

```
switch# configure terminal
switch(config)# 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 unreachable: false
```

Optional IPv6 Neighbor Discovery

You can use the following optional IPv6 Neighbor Discovery commands:

Table 7:

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.
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.

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

Table 8:

hardware ip verify address { destination zero identical reserved source multicast }	<p>Performs the following IDS checks on the IPv6 address:</p> <ul style="list-style-type: none"> • destination zero —Drops IPv6 packets if the destination IP address is ::. • identical —Drops IPv6 packets if the source IPv6 address is identical to the destination IPv6 address. • reserved —Drops IPv6 packets if the IPv6 address is ::1. • 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 }	<p>Performs the following IDS checks on the IPv6 address:</p> <ul style="list-style-type: none"> • consistent —Drops IPv6 packets where the Ethernet frame size is greater than or equal to the IPv6 packet length plus the Ethernet header. • maximum max-frag —Drops IPv6 packets if the formula (IPv6 Payload Length – IPv6 Extension Header Bytes) + (Fragment Offset * 8) is greater than 65536. • maximum max-tcp —Drops IPv6 packets if the TCP length is greater than the IP payload length. • 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.

Configuring IPv6 Stateless Autoconfiguration

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface management <i>number</i> Example: switch(config)# interface mgmt0	Specifies an interface type and number, and places the device in interface configuration mode.
Step 3	ipv6 address autoconfig Example: switch(config-if)# ipv6 address autoconfig	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on the management interface.
Step 4	ipv6 address autoconfig default Example: switch(config-if)# ipv6 address autoconfig default	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on the management interface and adds a default route with next-hop as that of the link-local address received in the router advertisement.

Example

This example shows how to use the `show ipv6 interface` command to display and verify that IPv6 addresses are configured on the management interface. Information displays the all the IPV6 addresses configured on the interface including the SLAAC generated addresses. It also indicates whether or not the stateless address autoconfig is enabled on the interface:

```
Device# show ipv6 interface mgmt 0

IPv6 Interface Status for VRF "management"(2)
mgmt0, Interface status: protocol-up/link-up/admin-up, iod: 2
IPv6 address:
1955::2f6:63ff:fe8b:c9f8/64 [VALID]
IPv6 subnet: 1955::/64
IPv6 link-local address: fe80::2f6:63ff:fe8b:c9f8 (default) [VALID]
....
Stateless autoconfig configured on the interface
```

This example shows how to use the `show ipv6 route vrf management` command to display the IPv6 routing table for VRF management:

```
Device# show ipv6 route vrf management

IPv6 Routing Table for VRF "management"
'*' denotes best ucast next-hop
 '**' denotes best mcast next-hop
 '[x/y]' denotes [preference/metric]
```

```

0::/0, ubest/mbest: 1/0
*via fe80::2f6:63ff:fe8b:c9ff, mgmt0, [2/0], 00:02:00, icmpv6
1955::/64, ubest/mbest: 1/0, attached
*via 1955::2f6:63ff:fe8b:c9f8, mgmt0, [0/0], 15:59:22, direct,
1955::2f6:63ff:fe8b:c9f8/128, ubest/mbest: 1/0, attached
*via 1955::2f6:63ff:fe8b:c9f8, mgmt0, [0/0], 15:59:22, local

```

This example shows how to use the `show ipv6 nd int mgmt` command to display the ICMPv6 ND interfaces for VRF management:

```

Device# show ipv6 nd int mgmt 0

ICMPv6 ND Interfaces for VRF "management"
mgmt0, Interface status: protocol-up/link-up/admin-up
IPv6 address:
1955::2f6:63ff:fe8b:c9f8/64 [VALID]
IPv6 link-local address: fe80::2f6:63ff:fe8b:c9f8 [VALID]
.....
Subnets configured via SLAAC and their states:
Prefix 1955::/64[PREFERRED] Preferred lifetime left: 6d23h Valid lifetime left: 4w1d

```

Verifying the IPv6 Configuration

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

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

```

Additional References

For additional information related to implementing IPv6, see the following sections:

Related Documents

Related Topic	Document Title
IPv6 CLI commands	Cisco Nexus 3000 Series NX-OS Unicast Routing Command Reference, Release 6.x

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	-

