Configuring IPv6

This chapter contains the following topics:

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, enables more efficient routing. IPv6 supports Routing Information Protocol (RIP), Integrated Intermediate System-to-Intermediate System (IS-IS), 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	2001:0:0:0:0:DB8:800:200C: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 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

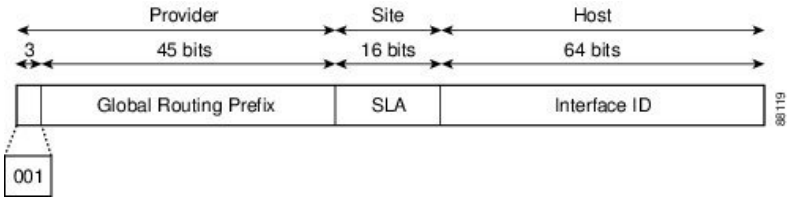
An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address.

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, 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:

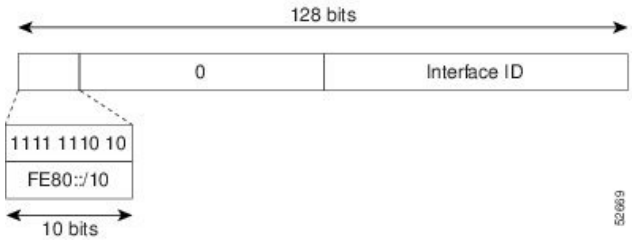
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses,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

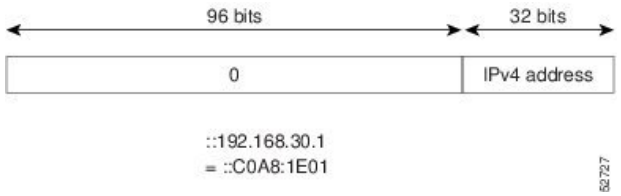
Figure 2: Link-Local Address Format



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 figure shows the structure of a n IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 3: IPv4-Compatible IPv6 Address Format



Unique Local Addresses

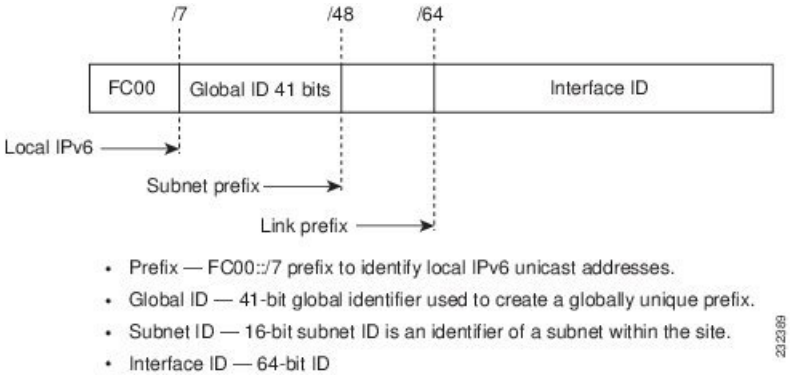
A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications might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

Figure 4: Unique Local Address Structure



Site Local Addresses

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.

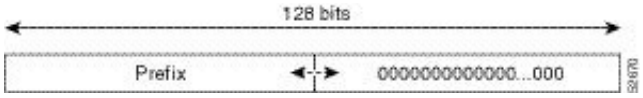
IPv6 Anycast Addresses

An anycast address is an address that is assigned to a set of interfaces that belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface turns a unicast address into an anycast address. You must configure the nodes to which the anycast address belongs to recognize that the address is an anycast address.

Note Anycast addresses can be used only by a router, not a host. Anycast addresses cannot be used as the source address of an IPv6 packet.

The following figure shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

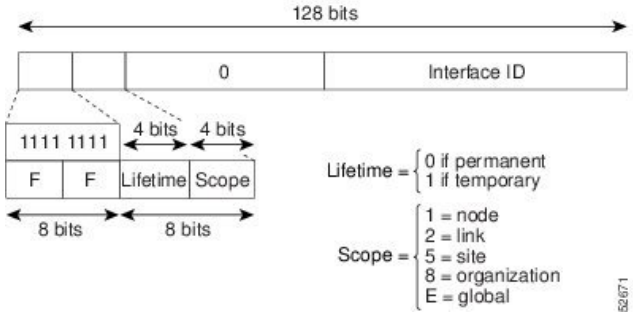
Figure 5: Subnet Router Anycast Address Format



IPv6 Multicast Addresses

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope, has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. The following figure shows the format of the IPv6 multicast address.

Figure 6: IPv6 Multicast Address Format



IPv6 nodes (hosts and routers) are required to join (where received packets are destined for) the following multicast groups:

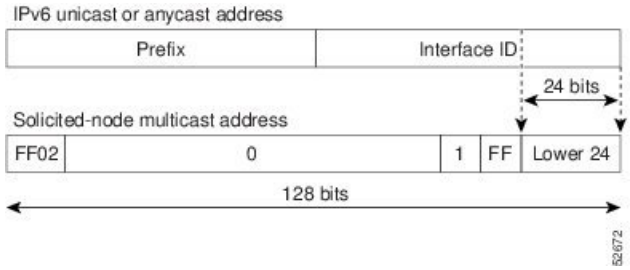
- All-nodes multicast group FF02:0:0:0:0:0:0:1 (the scope is link-local)

- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (the scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which they are assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address (see the figure below). For example, the solicited-node multicast address that corresponds to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

Figure 7: IPv6 Solicited-Node Multicast Address Format

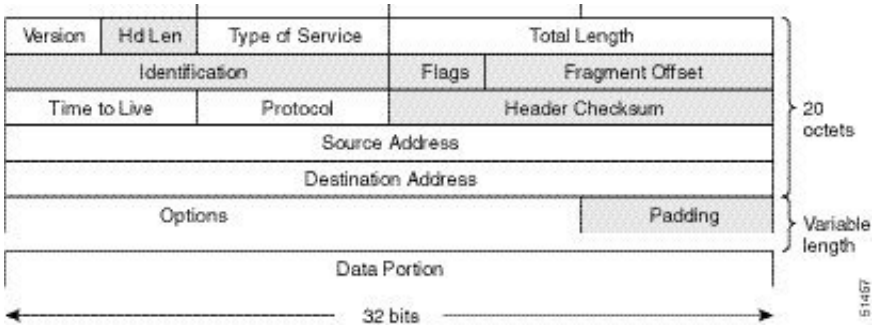


Note IPv6 has no broadcast addresses. IPv6 multicast addresses are used instead of broadcast addresses.

IPv4 Packet Header

The base IPv4 packet header has 12 fields with a total size of 20 octets (160 bits). 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 8: IPv4 Packet Header Format



Simplified IPv6 Packet Header

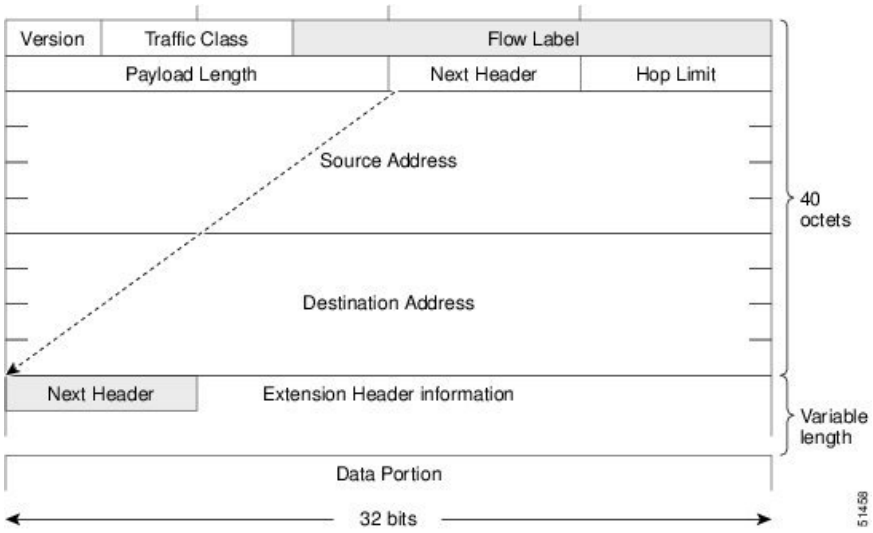
The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). Fragmentation is handled by the source of a packet, and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet, and the base IPv6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

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

Table 2: Base IPv6 Packet Header Fields

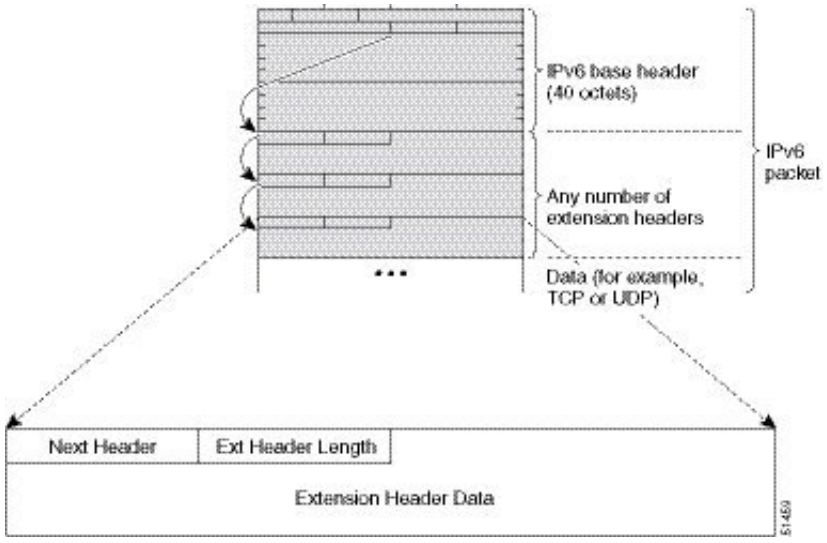
Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is 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.
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 9: 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 10: 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 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.
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) 17 (UDP)	Headers that are used inside a packet to transport the data. The two main transport protocols are TCP and UDP.

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 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.c0.yyyy.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.

LPM Routing Modes for IPv6

By default, Cisco NX-OS programs routes in a hierarchical fashion to allow for the longest prefix match (LPM) on the device. However, you can configure the device for different routing modes to support more LPM route entries.

The following tables list the LPM routing modes that are supported on Cisco Nexus 3400-S platform switches.

Table 5: LPM Routing Modes for Cisco Nexus 3400-S Platform Switches

LPM Routing Mode	CLI Command
Default system routing mode	
LPM heavy routing mode	**system routing template-lpm-heavy**

Neighbor Discovery Local Proxy

This feature makes the switch respond to all Neighbor Solicitation (NS) packets on the configured subnet. Even in the case for hosts that normally do not need routing.

The Neighbor Discovery (ND) Local Proxy feature is enabled by using the following command: **ipv6 nd local-proxy** *prefix* **no-hw-flooding**.

In the case of IPv6, multiple subnets can be configured on the same interface. Because of this, ND Local Proxy has to be turned on per subnet on an interface having multiple IPv6 addresses.

For subnets for which ND Local Proxy is enabled, in the case of multicast NS packets, the switch proxies for the host to which the multicast NS is destined with its own gateway MAC.

In case of an interface having multiple subnets, for subnets for which ND Local Proxy is not enabled, the switch does not proxy for the hosts in that subnet. The switch looks up the target address in the received multicast NS packet. If it is there in a subnet for which ND Local Proxy is not enabled, it is dropped.

For Unicast NS packets:

- The switch proxies for hosts in those subnets and for which a proxy is enabled.

- The switch does not proxy for other subnets for which proxy is not enabled.

Multicast NS packets for Link Local address of hosts: If the interface has one or more subnets for which ND Local Proxy is enabled, multicast packets destined to the Solicited Node Multicast address (derived from the Link Local address of hosts), the switch floods back the NS resulting in the host getting the multicast NS packet. There is no proxy in this case.

The following applies to subnet proxy:

- Multiple IPv6 subnets can be configured on an interface as there is the concept of a secondary IP address in IPv6.
- The **ipv6 nd local-proxy** *network-prefix/mask-len* **no-hw-flooding** command has been added.

Virtualization Support

IPv6 supports virtual routing and forwarding (VRF) instances.

Prerequisites for IPv6

IPv6 has the following prerequisites:

- You must be familiar with IPv6 basics such as IPv6 addressing and IPv6 header information.
- 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. You can directly attach IPv6 hosts 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.
- IPv6 static route next-hop link-local addresses cannot be configured at any local interface.
- You must define the BGP update source when using a link-local IPv6 address.
- Because RFC 3879 deprecates the use of site-local addresses, configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.
- When a better route to a destination exists on the same interface, packets are not sent to the supervisor module (SUP) and ICMP redirection does not occur. Therefore, packets can take sub-optimal paths to their destinations.

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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ipv6 address** {*address* [**eui64**] [**route-preference** *preference*] [**secondary**] [**tag** *tag-id*] or **ipv6 address** *ipv6-address* **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:**	Saves this configuration change.

	Command or Action	Purpose
	`switch(config-if)# copy running-config startup-config`	

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 LPM Heavy Routing Mode

You can configure LPM heavy routing mode in order to support more LPM route entries.

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

Note For LPM heavy routing mode scale numbers, see the *Cisco Nexus 3400-S NX-OS Verified Scalability Guide*.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **system routing template-lpm-heavy** **Example:** `switch(config)# system routing template-lpm-heavy`	Puts the device in LPM heavy routing mode to support a larger LPM scale.

Step 3	(Optional) **show system routing mode**
	Example:
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: LPM Heavy`
	`Applied System Routing Mode: LPM Heavy`
		Displays the LPM routing mode.

Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.
Step 5	**reload** **Example:** `switch(config)# reload`	Reboots the entire device.

Configuring IPv6 ND Local Proxy on SVIs

You can configure local proxy ND on SVIs along with local proxy ARP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface vlan** *vlan-id*
	Example:
	`switch(config)# interface vlan 1002`
	`switch(config-if)#`
		Creates a VLAN interface and enters the configuration mode for the SVI.

	Command or Action	Purpose
Step 3	**ipv6 nd local-proxy** *address/mask* **no-hw-flooding**] **Example:** `switch(config-if)# ip nd local-proxy 1::1/64 no-hw-flooding`	Enables local proxy ND for all subnets configured on the SVI.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the IPv6 Configuration

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

Command	Purpose
show ipv6 interface	Displays IPv6-related interface information.
show ipv6 adjacency	Displays the adjacency table.
show system routing mode	Displays the LPM routing mode.

Configuration Examples for IPv6

The following example shows how to configure IPv6:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 address 2001:db8::/64 eui64
switch(config-if)# ipv6 nd reachable-time 10
```