

IPv6 Addresses

This chapter contains theses topics:

- IPv6 addresses, on page 1
- Virtualization support, on page 22
- IPv6 routes with ECMP, on page 23
- Prerequisites for IPv6, on page 23
- Guidelines and limitations for IPv6, on page 23
- Configure IPv6, on page 26
- Verify the IPv6 configuration, on page 44
- Verify the IPv6 configuration, on page 44

IPv6 addresses

IPv6 addresses are a network protocol designed to replace IPv4 addresses by expanding the address space and improving network functionality. IPv6 increases the address size from 32 bits in IPv4 to 128 bits, allowing for a vastly larger number of unique addresses. This expansion supports network scalability and global reachability.

Key attributes of IPv6 addresses include:

- A much larger address space that reduces the need for private addresses and Network Address Translation (NAT).
- A simplified main header and the use of extension headers, which improve packet processing efficiency.
- Support for new application protocols that operate without requiring special processing by border routers.
- Enhanced routing capabilities through features like prefix aggregation, simplified network renumbering, and site multihoming.
- Compatibility with routing protocols such as 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).

These attributes make IPv6 a more flexible and efficient protocol for modern and future network environments, enabling better routing, easier network management, and support for emerging applications.

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

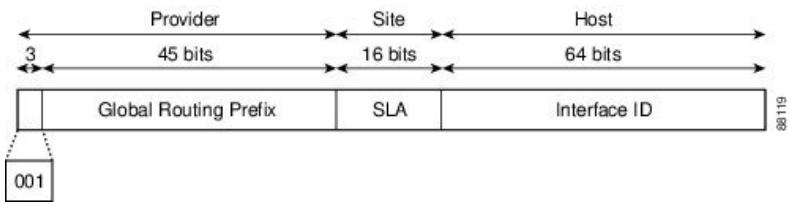
An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix that

- enables strict aggregation of routing prefixes,
- limits the number of routing table entries in the global routing table, and
- is used on links aggregated upward through organizations to Internet service providers (ISPs).

This structure supports efficient routing by allowing hierarchical aggregation of addresses.

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. All global unicast addresses, except those that start with binary 000, 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). See Figure 1: Aggregatable Global Address Format , on page 3 for 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 is a 16-bit subnet field that organizations can use to create a local addressing hierarchy and identify subnets. A subnet ID is similar to a subnet in IPv4. 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.

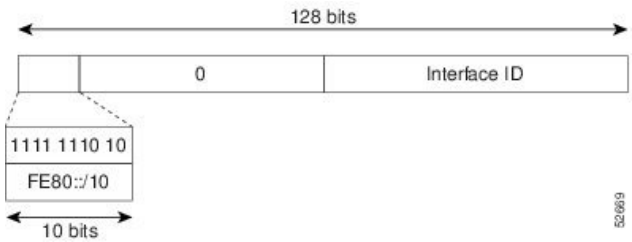
Link-Local addresses

A link-local address is an IPv6 unicast address. It 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. These nodes do not need globally unique addresses to communicate. Figure Figure 1: Aggregatable Global Address Format , on page 3 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

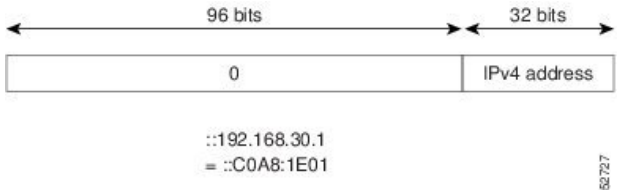


IPv4-Compatible IPv6 addresses

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. Figure Figure 1: Aggregatable Global Address Format , on page 3 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.

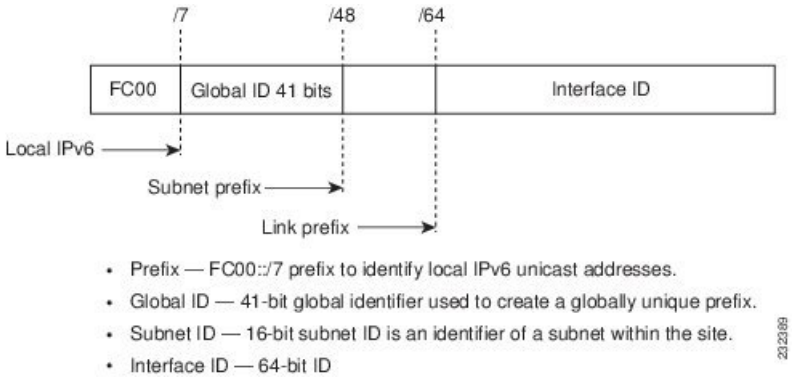
Unique local addresses are not expected to be routed on the global Internet. It can be routed inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications might treat unique local addresses like global scoped addresses.

These are the characteristics of unique local address:

- 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.

Figure Figure 1: Aggregatable Global Address Format , on page 3shows the structure of a unique local address.

Figure 4: Unique Local Address Structure



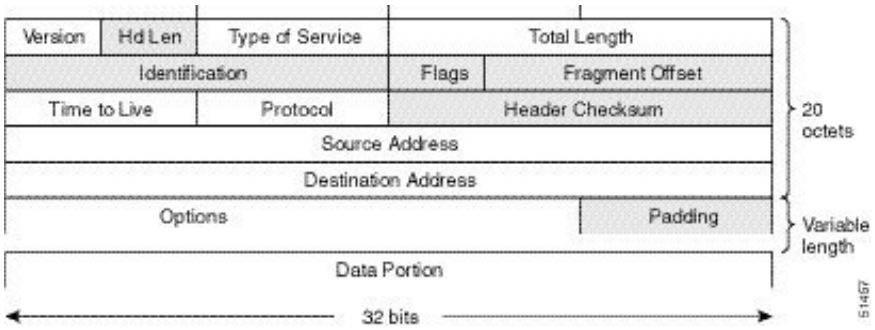
Site local addresses

Because RFC 3879 deprecates the use of site-local addresses, 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). 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 headers

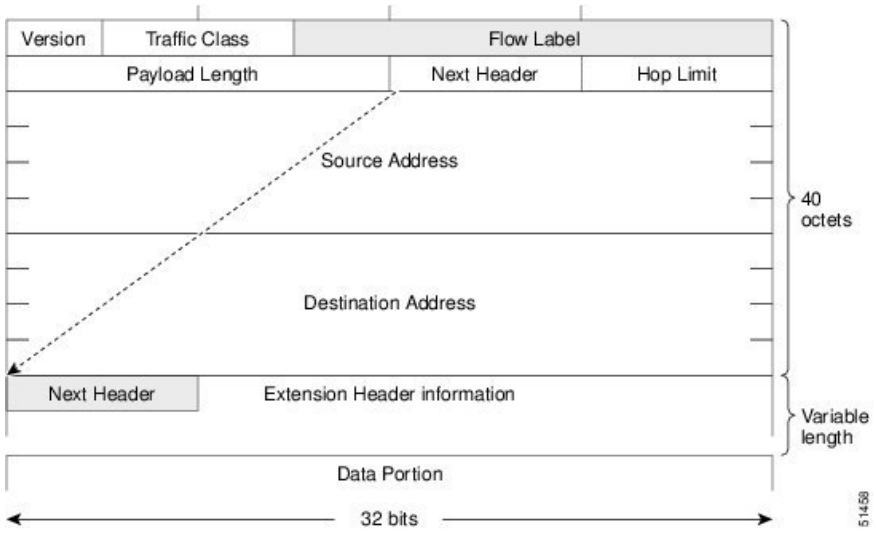
The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). Fragmentation is handled by the source of a packet, and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet, and the base IPv6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

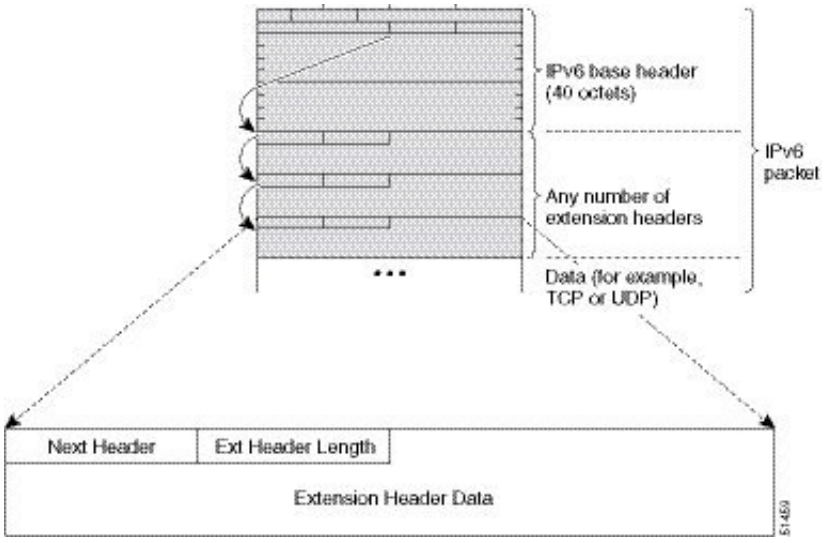
Figure 6: IPv6 Packet Header Format



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 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.
Shim6	140	Header that is used for IP multihoming, which allows a host to be connected to multiple networks.
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.

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 9000 Series 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. See Table 4: IPv6 DNS Record Types, on page 10 table for the DNS record types support IPv6 addresses.

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 maximum transmission unit (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

Cisco Discovery Protocol (CDP) IPv6 address support is a feature that

- transfers IPv6 addressing information between directly connected Cisco devices,
- provides IPv6 neighbor information to network management and troubleshooting tools, and
- enhances network visibility and management for IPv6-enabled environments.

ICMP for IPv6

Internet Control Message Protocol for IPv6 (ICMPv6) is a network-layer protocol that

- works with IPv6
- reports errors if packets cannot be processed correctly, and
- sends informational messages about the status of the network.

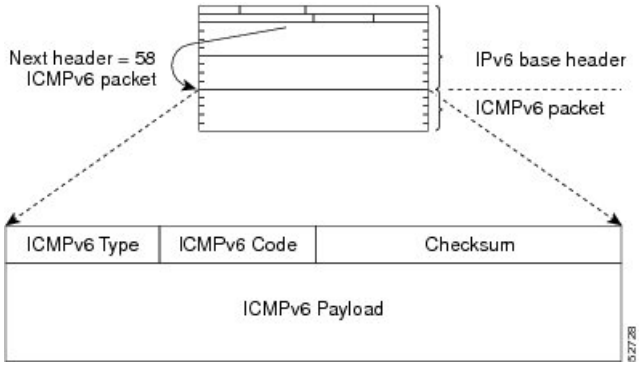
When a router is unable to forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures 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. Figure Figure 8: IPv6 ICMP Packet Header Format , on page 12 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 NDP to

- determine the addresses of nodes on the same network (local link)
- find neighboring routers that can forward their packets
- verify whether neighboring routers are reachable or not, and
- to detect changes to link-layer addresses.

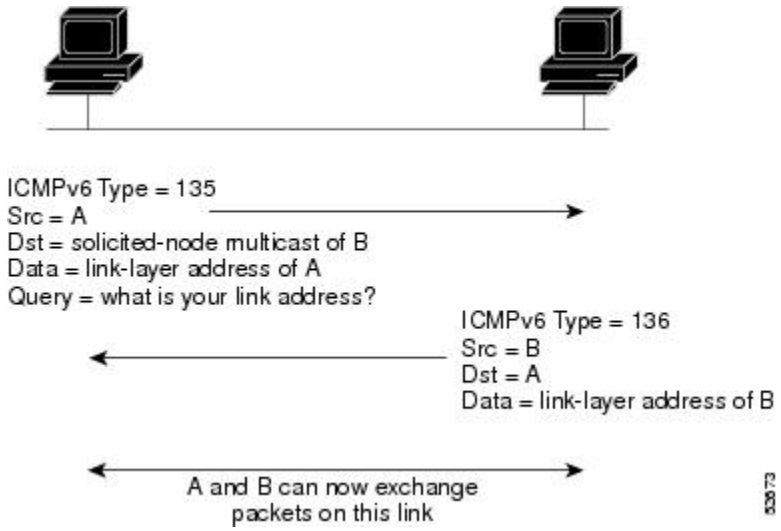
NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

IPv6 neighbor solicitation messages

IPv6 neighbor solicitation messages are part of the Neighbor Discovery Protocol used by nodes on the same local link to determine each other's link-layer addresses and verify reachability.

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. 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. See Figure 9: IPv6 Neighbor Discovery-Neighbor Solicitation Message , on page 13 for more details.

Figure 9: IPv6 Neighbor Discovery-Neighbor Solicitation Message



A neighbor solicitation message is an ICMPv6 packet with a Type field value of 135. A node sends this message on the local link when it needs to find the link-layer address of another node on the same link. The message includes:

- The source address: the IPv6 address of the sending node.
- The destination address: the solicited-node multicast address corresponding to the target node's IPv6 address.
- The link-layer address of the source node.

When the destination node receives a neighbor solicitation, it replies with a neighbor advertisement message, which has a Type field value of 136. This message includes:

- The source address: the IPv6 address of the node sending the advertisement.
- The destination address: the IPv6 address of the node that sent the solicitation.
- The link-layer address of the advertising node.

After the source node receives this advertisement, the two nodes can communicate directly.

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 autoconfigurations

IPv6 Stateless Autoconfigurations are mechanisms that enable IPv6 nodes to automatically configure their own IP addresses without requiring manual setup or a DHCP server. These autoconfigurations ensure that every interface on an IPv6 node has a link-local address, which is essential for communication on the local network segment.

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 compute node IP autoconfigurations

An IPv6 Compute Node IP Auto-Configuration is a feature that

- automatically assigns unique IPv6 node IP addresses to connected multi-homed compute nodes,
- enables these nodes to be onboarded into a Kubernetes (K8s) cluster, and
- establishes eBGP peering between the switch and each compute node.

A node IP must be assigned to connected compute nodes before they can be on-boarded into a K8s cluster and eBGP peering can be established between the switch and a compute node.

Beginning with Cisco NX-OS Release 10.3(3)F, the IPv6 Compute Node IP Auto-Configuration support is provided on Cisco NX-OS 9000 series platform switches to assign and distribute the node IP addresses to multi-homed compute nodes and establish reachability to K8s cluster using the assigned node IP.



Note The node address assignment is however different from SLAAC. It is a method to assign an unique IPv6 address on the loopback interface that is orthogonal to interface address provisioning in a layer-3 interface subnet that is done through SLAAC.

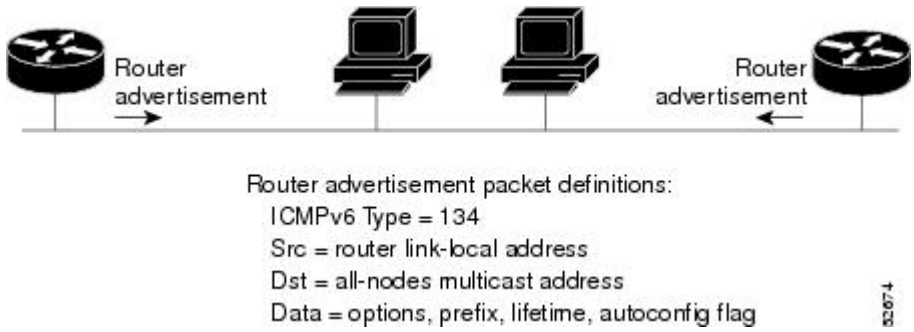
This feature complies with the standard as defined in RFC 8505/6775.

IPv6 router advertisement messages

IPv6 Router Advertisement (RA) messages are ICMP packets with a type field value of 134 that routers periodically send out on each configured interface. These messages enable IPv6 stateless6 stateless address autoconfiguration by advertising network prefixes and other configuration parameters to hosts on the local link.

The RA messages are sent to all-nodes multicast address as shown in the figure Figure 9: IPv6 Neighbor Discovery-Neighbor Solicitation Message , on page 13.

Figure 10: IPv6 Neighbor Discovery-RA Message



RA messages typically include these 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 these RA message parameters:

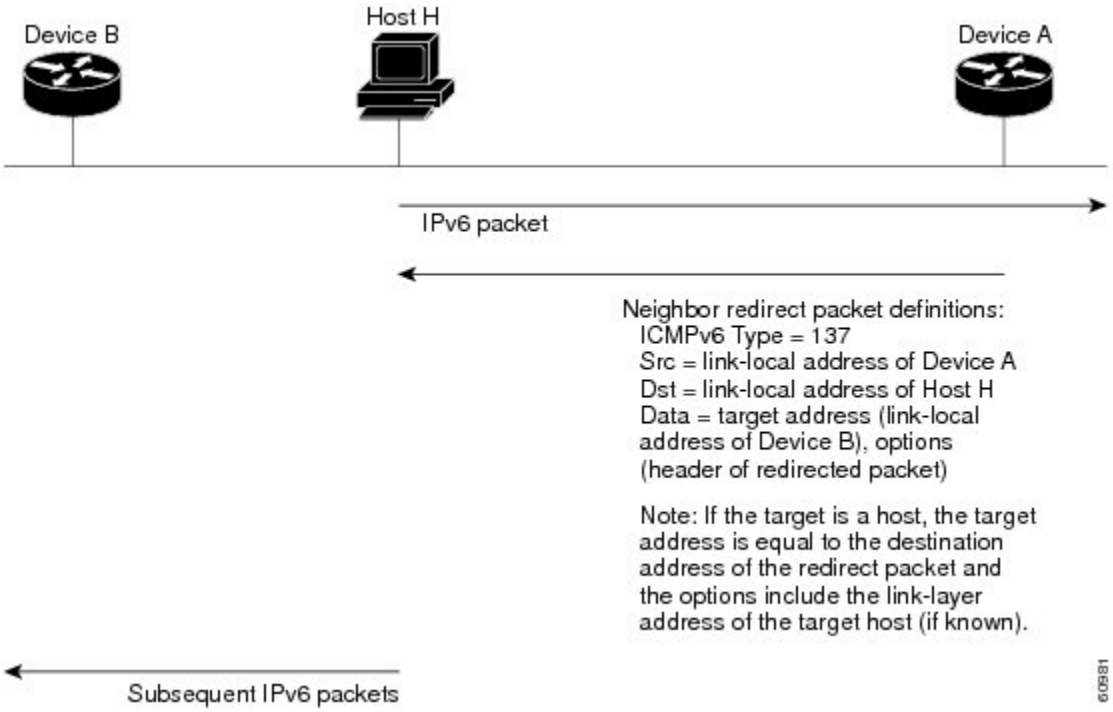
- The time interval between periodic RA messages.
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link).
- The network prefixes in use on a given link.
- The time interval between neighbor solicitation message retransmissions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

An IPv6 neighbor redirect message is a type of Internet Control Message Protocol (ICMP) packet that routers send to hosts to inform them of a better first-hop node on the path to a destination. This message is identified by a value of 137 in the Type field of the ICMP packet header.

Figure 11: IPv6 Neighbor Discovery-Neighbor Redirect Message



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 theses 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.

IPv6 anycast addresses

An anycast address is an address that is assigned to a set of interfaces that belong to different nodes.

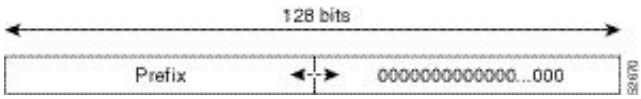
A packet sent to an anycast address is delivered to the closest interface as defined by the routing protocols in use identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space.

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 by a host. Anycast addresses cannot be used as the source address of an IPv6 packet.

Figure Figure 12: Subnet Router Anycast Address Format , on page 18 represents 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 12: Subnet Router Anycast Address Format



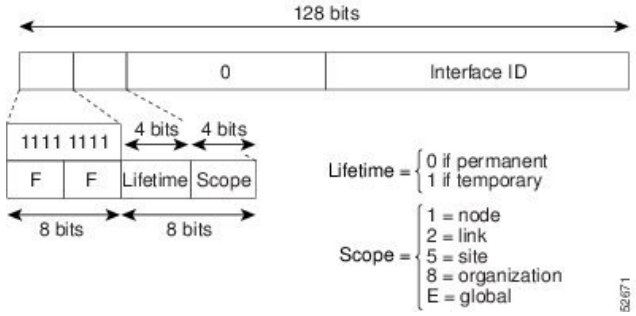
IPv6 multicast addresses

IPv6 multicast addresses are a type of IPv6 address that identify a group of interfaces across different nodes. These addresses always start with the prefix FF00::/8 (binary 1111 1111). When a packet is sent to an IPv6 multicast address, it is delivered to all interfaces that belong to the multicast group.

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. Figure Figure 12: Subnet Router Anycast Address Format , on page 18 shows the format of the IPv6 multicast address.

Figure 13: IPv6 Multicast Address Format



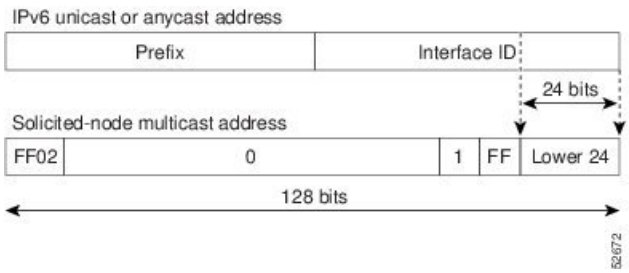
IPv6 nodes (hosts and routers) are required to join (where received packets are destined for) these multicast groups:

- 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 14: IPv6 Solicited-Node Multicast Address Format



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

LPM routing modes

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.

Theses tables list the LPM routing modes that are supported on Cisco Nexus 9000 Series switches.

LPM routing modes for Cisco Nexus 9200 platform switches

Table 5: LPM routing modes for Cisco Nexus 9200 platform switches

LPM routing mode	CLI command
Default system routing mode	
LPM dual-host routing mode	**system routing template-dual-stack-host-scale**
LPM heavy routing mode	**system routing template-lpm-heavy**

Note Cisco Nexus 9200 platform switches do not support the **system routing template-lpm-heavy** mode for IPv4 Multicast routes. Make sure to reset LPM's maximum limit to 0.

LPM routing modes for Cisco Nexus 9300 platform switches

Table 6: LPM routing modes for Cisco Nexus 9300 platform switches

LPM routing mode	Broadcom T2 mode	CLI command
Default system routing mode	3	
ALPM routing mode	4	**system routing max-mode l3**

LPM routing modes for Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches

Table 7: LPM routing modes for Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches

LPM routing mode	CLI command
LPM dual-host routing mode	**system routing template-dual-stack-host-scale**
LPM heavy routing mode	**system routing template-lpm-heavy**
LPM Internet-peering mode	**system routing template-internet-peering**

LPM routing modes for Cisco Nexus 9300-FX platform switches

LPM routing modes for Cisco Nexus 9300-fx2 platform switches

LPM routing modes for Cisco Nexus 9300-GX platform switches

LPM routing modes for Cisco Nexus 9500 platform switches with 9700-ex and 9700-fx line cards

Table 8: LPM routing modes for Cisco Nexus 9500 platform switches with 9700-ex and 9700-fx line cards

LPM routing mode	Broadcom T2 mode	CLI command
Default system routing mode	3 (for line cards); 4 (for fabric modules)	
Max-host routing mode	2 (for line cards); 3 (for fabric modules)	**system routing max-mode host**
Nonhierarchical routing mode	3 (for line cards); 4 with max-l3-mode option (for line cards)	**system routing non-hierarchical-routing [max-l3-mode]**
64-bit ALPM routing mode	Submode of mode 4 (for fabric modules)	**system routing mode hierarchical 64b-alpm**
LPM heavy routing mode		**system routing template-lpm-heavy** **Note** This mode is supported only for Cisco Nexus 9508 switches with the 9732C-EX line card.

LPM routing mode	Broadcom T2 mode	CLI command
LPM Internet-peering mode		**system routing template-internet-peering** **Note** This mode is supported only for the following Cisco Nexus 9500 Platform Switches: • Cisco Nexus 9500 platform switches with 9700-EX line cards. • Cisco Nexus 9500-FX platform switches (Cisco NX-OS release 7.0(3)I7(4) and later) • Cisco 9500-R platform switches (Cisco NX-OS release 9.3(1) and later)
LPM dual-host routing mode		

LPM routing modes for Cisco Nexus 9500-R platform switches with 9600-R line cards

Table 9: LPM Routing Modes for Cisco Nexus 9500-R Platform Switches with 9600-R Line Cards

LPM routing Mode	CLI command
LPM Internet-peering mode	**system routing template-internet-peering** (Cisco NX-OS release 9.3(1) and later)

Host to LPM spillover

Beginning with Cisco NX-OS Release 7.0(3)I5(1), host routes can be stored in the LPM table in order to achieve a larger host scale. In ALPM mode, the switch allows fewer host routes. If you add more host routes than the supported scale, the routes that are spilled over from the host table take the space of the LPM routes in the LPM table. The total number of LPM routes allowed in that mode is reduced by the number of host routes stored. This feature is supported on Cisco Nexus 9300 and 9500 platform switches.

In the default system routing mode, Cisco Nexus 9300 platform switches are configured for higher host scale and fewer LPM routes, and the LPM space can be used to store more host routes. For Cisco Nexus 9500 platform switches, only the default system routing and nonhierarchical routing modes support this feature on line cards. Fabric modules do not support this feature.

Virtualization support

IPv6 supports virtual routing and forwarding (VRF) instances.

IPv6 routes with ECMP

IPv6 routes with Equal-Cost Multipath (ECMP) are routing entries that distribute traffic across multiple next-hops with equal cost to optimize network performance and redundancy.

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

These are the prerequisites of IPv6:

- 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 these configuration guidelines and limitations:

- Cisco Nexus 9300-EX and Cisco Nexus 9300-FX2 platform switches configured for internet-peering mode might not have sufficient hardware capacity to install full IPv4 and IPv6 Internet routes simultaneously.
- 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.
- Usage of IPv6 LLA requires the TCAM Region for **ing-sup** to be re-carved from the default value of 512 to 768. This step requires a copy run start and reload
- 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, you should configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.
- When **system routing template-internet-peering** is configured, multicast traffic is not supported. However, it is still possible to have feature PIM and PIM configurations in the switch.

- For Cisco Nexus 9500-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/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.
- LPM heavy routing mode is supported on Cisco Nexus 9500 series switches with 9700 -EX, -FX, and -GX series modules.
- Beginning with Cisco NX-OS Release 10.2(3)F, syslog will be printed when IPv6 redirect message is triggered based on the configured interval.
- Beginning with Cisco NX-OS Release 10.3(1)F, static routing is supported on the Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, static routing is supported on the Cisco Nexus 9804 switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, dynamic routing is supported on the Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, dynamic routing is supported on the Cisco Nexus 9804 switches.
- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 Compute Node IP Auto-Configuration feature is supported on Cisco NX-OS 9000 series platform switches with the following limitations:
  - The RA prefix must be configured as offlink, with the prefix length of 64.
  - If there is a multi-homed compute node, same RA prefix must be configured on both L1 and L2 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, dynamic routing is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with 9808 and 9804 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, static routing is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with 9808 and 9804 switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, static routing is supported on Cisco Nexus 9232E-B1 switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, the following IPv6 Layer-3 features are supported on the Cisco Nexus 9232E-B1 switches:
  - IPv6 unicast routes
  - IPv6 host routes
  - IPv6 Neighbor Discovery
  - BGP Neighbors
- Beginning with Cisco NX-OS Release 10.6(1)F, the following IPv6 Layer-3 features are supported on the Cisco Nexus N9336C-SE1 switches:
  - IPv6 unicast routes

- IPv6 host routes
- IPv6 Neighbor Discovery
- ARP

- Beginning with Cisco NX-OS Release 10.6(2)F, you can use **template-l3-host-heavy** to handle /128 host routes and increased IPv6 route scale up to the verified limit. This new template is introduced on Cisco Nexus N9300-FX2 platform switches.
- Beginning with Cisco NX-OS Release 10.6(2)F, IPv6 is supported on Cisco Nexus N9324C-SE1U and N9348Y2C6D-SE1U switches.
- Beginning with Cisco NX-OS Release 10.6(3)F, you can use **template-l3-host-heavy** on Cisco Nexus N9K-C9364C-H1, N9K-C9332D-H2R, and N9K-C93400LD-H1 platform switches.
- Beginning with Cisco NX-OS Release 10.6(2n), Layer 3 support for IPv6 is supported on the Cisco N9164E-NS4-O switch.
- Beginning with Cisco NX-OS Release 10.6(3)F, you can configure the neighbour discovery cache limit at the SVI level. This feature improves resource optimization, security, and network performance.
  - This feature enables the restriction on at the software level. The ND packets are still copied to slow path even after reaching limits.
  - You can enable this feature in interface mode as well as in global configuration mode. However, interface level configuration takes precedence over the global configuration mode.
  - This feature is supported on SVI and not supported on these interfaces:
    - Ethernet
    - Sub-interfaces
    - Port-channel
    - Un-numbered Interfaces
  - If users enable this feature in non-supporting interface mode such as management or ethernet interface, the configuration is applied in global configuration mode and this warning message is shown:

    ```
    Warning: ND cache limit configured globally. If ND cache count already exceeds the
    configured value, excessive entries will be removed in next refresh cycle.
    ```

  - Remove this configuration before downgrading to the releases earlier than NX-OS 10.6(3)F where this feature is not supported.

Configure IPv6

Configure IPv6 addresses

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

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **interface ethernet** *number* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 2/3
switch(config-if)#
```

Step 3 Us the **ipv6 address** {*address* [**eui64**] [**route-preference** *preference*] [**secondary**] [**tag** *tag-id*] or **ipv6 address** *ipv6-address* **use-link-local-only** command to specify an IPv6 address assigned to the interface and enable IPv6 processing on the interface.

Example:

```
switch(config-if)# ipv6 address
2001:0DB8::1/10
```

or

```
switch(config-if)# ipv6 address
use-link-local-only
```

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) Use the **show ipv6 interface** command to see interfaces configured for IPv6.

Example:

```
switch(config-if)# show ipv6 interface
```

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-if)# copy running-config
startup-config
```

Example

This example shows how to configure an IPv6 address:

```
switch# <userinput>configure terminal</userinput>
    switch(config)# <userinput>interface ethernet 3/1</userinput>
    switch(config-if)# <userinput>ipv6 address ?</userinput>
    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)# <userinput>ipv6 address 2001:db8::/64 eui64</userinput>
```

This example shows how to display an IPv6 interface:

```
switch(config-if)# <userinput>show ipv6 interface ethernet 3/1</userinput>
    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
```

Configure max-host routing mode (Cisco Nexus 9500 Platform switches only)

By default, the device programs routes in a hierarchical fashion (with fabric modules that are configured to be in mode 4 and line card modules that are 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.

Note If you want to further scale the entries in the LPM table, see the Configuring Nonhierarchical Routing Mode (Cisco Nexus 9500 Series Switches Only) section to configure the device to program all the Layer 3 IPv4 and IPv6 routes on the line cards and none of the routes on the fabric modules.

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

Note For the max-host routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the [**no**] **system routing max-mode host** command to put the line cards in Broadcom T2 mode 2 and the fabric modules in Broadcom T2 mode 3 to increase the number of supported hosts.

Example:

```
switch(config)# system routing max-mode host
```

Step 3 (Optional) Use the **show forwarding route summary** command to see see the LPM routing mode.

Example:

```
switch(config)# show forwarding route summary
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure nonhierarchical routing mode (Cisco Nexus 9500 series switches only)

If the host scale is small (as in a pure Layer 3 deployment), we recommend programming the longest prefix match (LPM) routes in the line cards to improve convergence performance. Doing so programs routes and hosts in the line cards and does not program any routes in the fabric modules.

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

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Use the [**no**] **system routing non-hierarchical-routing** [**max-l3-mode**] to put the line cards in Broadcom T2 mode 3 (or Broadcom T2 mode 4 if you use the **max-l3-mode** option) to support a larger LPM scale.

Example:

```
switch(config)# system routing
 non-hierarchical-routing max-l3-mode
```

As a result, all of the IPv4 and IPv6 routes will be programmed on the line cards rather than on the fabric modules.

Step 3 (Optional) Use the **show forwarding route summary** command to see the LPM mode.

Example:

```
switch(config)# show forwarding route
                        summary
                        Mode 3:
                        120K IPv4 Host table
                        16k LPM table (> 65 < 127 1k entry
                        reserved)
                        Mode 4:
                        16k V4 host/4k V6 host
                        128k v4 LPM/20K V6 LPM
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config
startup-config
```

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure 64-Bit ALPM routing mode (Cisco Nexus 9500 platform switches only)

You can use the 64-bit algorithmic longest prefix match (ALPM) feature to manage IPv4 and IPv6 route table entries. In 64-bit ALPM routing mode, the device can store more route entries. In this mode, you can program one of the following:

- 80,000 IPv6 entries and no IPv4 entries
- No IPv6 entries and 128,000 IPv4 entries
- *x* IPv6 entries and y IPv4 entries, where $2x + y <= 128,000$

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

Note For the 64-bit ALPM routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 use the [**no**] **system routing mode hierarchical 64b-alpm** command to program all IPv4 and IPv6 LPM routes with a mask length less than or equal to 64 in the fabric module.

Example:

```
switch(config)# system routing mode
hierarchical 64b-alpm
```

All host routes for IPv4 and IPv6 and all LPM routes with a mask length of 65–127 are programmed in the line card.

Step 3 (Optional) Use the **show forwarding route summary** command to see the LPM mode.

Example:

```
switch(config)# show forwarding route
summary
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config
startup-config
```

Saves this configuration change.

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure ALPM routing mode (Cisco Nexus 9300 platform switches only)

You can configure Cisco Nexus 9300 platform switches to support more LPM route entries.

This configuration impacts both the IPv4 and IPv6 address families.

- For ALPM routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the [**no**] **system routing max-mode l3** command to put the device in Broadcom T2 mode 4 to support a larger LPM scale.

Example:

```
switch(config)# system routing max-mode l3
```

Step 3 (Optional) Use the **show forwarding route summary** command to see the LPM mode.

Example:

```
switch(config)# show forwarding
route summary
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config
startup-config
```

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure 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

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **interface ethernet** *number* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 2/3
                        switch(config-if)#
```

Step 3 Use the **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**] command to specify an IPv6 address assigned to the interface and enables IPv6 processing on the interface.

Example:

```
switch(config-if)# ipv6 nd prefix
```

- **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.



Note When disabling IPv6 redirects, IPv4 redirects should also be disabled as some IPv6 packets may still be leaked to the CPU.

- **retrans-timer**— time-Advertises the time between neighbor-solicitation messages in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds.

- **suppress-ra**— Disables sending ICMPv6 router-advertisement messages.

Step 4 (Optional) Use the **show ip nd interface** command to see interfaces configured for IPv6 neighbor discovery.

Example:

```
switch(config-if)# show ip interface
```

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-if)# copy running-config startup-config
```

Example

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

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

This example shows how to display an IPv6 interface:

```
switch# <userinput>configure terminal</userinput>
               switch(config)# <userinput>show ipv6 nd interface ethernet 3/1</userinput>
               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:

Table 10:

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.

Configure IPv6 ND cache limit at the SVI

You can configure ND cache limit at the global configuration level as well as in interface level.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal

switch(config)#
```

-

Step 2 Use the **[no] ipv6 nd cache intf-limit** *limit* command to configure ND cache limit at the global configuration level.

Example:

```
switch(config)# ipv6 nd cache intf-limit 10

Range: 1-98000
```

Use the **no** form of this command to remove the configuration.

When you configure this CLI, ICMP generates this syslog warning message.



Warning ND cache limit configured globally. If ND cache count already exceeds the configured value, excessive entries will be removed in next refresh cycle.

-

Step 3 Use the **interface vlan** *vlan* command to enter SVI configuration mode.

Example:

```
switch(config)# interface vlan 10

switch(config-if)
```

-

Step 4 Use the **no] ipv6 nd cache intf-limit limit** *limit* command to configure ND cache limit at the SVI level.

Example:

```
switch(config-if)# ipv6 nd cache intf-limit 10

Range: 1-98000
```

Use the **no** form of this command to remove the configuration.

SVI level configuration takes precedence over the global configuration mode.

If the configured ND cache limit per interface is less than the currently existing total number of dynamically resolved ND entries for a given interface, ICMPv6 generates this syslog message:

```
Current number of ND cache entries(#total-count) exceeds the configured limit(#limit) for interface
(#intf-name) - Excessive entries will be removed in next refresh cycle.
```

After configuring, when ND cache count reaches the configured limit, ICMPv6 generates this syslog message:

```
Number of ND cache entries exceeded configured limit (#limit) for interface #Interface-name
```

-

Configure 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.

Procedure

Step 1 Enable IPv6 IDS checks in global configuration mode using the following commands as needed:

Example:

```
hardware ip verify address {destination zero | identical | reserved | source multicast}
                       hardware ipv6 verify length {consistent | maximum {max-frag | max-tcp | udp}}

                        hardware ipv6 verify tcp tiny-frag
                        hardware ipv6 verify version
```

The following commands perform various IDS checks on IPv6 packets:

Table 11: IPv6 IDS Command Reference

Command	Description
hardware ip verify address {**destination zero** | **identical** | **reserved** | **source multicast** }	Performs the following IDS checks on the IPv6 address: • **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** }}	Performs the following IDS checks on the IPv6 address: • **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).

The IDS checks are enabled for IPv6 packet verification as configured.

Step 2 Display the IPv6 packet verification configuration.

Example:

```
show hardware forwarding ip verify
```

The current IPv6 packet verification configuration is displayed.

Configure IPv6 stateless autoconfiguration

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Use the **interface management** *number* command to specify an interface type and number, and place the device in interface configuration mode.

Example:

```
switch(config)# interface mgmt0
```

Step 3 Use the **ipv6 address autoconfig** command to enable automatic configuration of IPv6 addresses using stateless autoconfiguration on the management interface.

Example:

```
switch(config-if)# ipv6 address autoconfig
```

Step 4 Use the **ipv6 address autoconfig default** command to enable 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:

```
switch(config-if)# ipv6 address autoconfig default
```

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
lef
                t: 4w1d
```

Configure LPM heavy routing mode (Cisco Nexus 9200 and 9300-EX platform switches and 9732C-EX Line card only)

Beginning with Cisco NX-OS Release 7.0(3)I4(4), you can configure LPM heavy routing mode in order to support significantly more LPM route entries. Only the Cisco Nexus 9200 and 9300-EX Series switches and the Cisco Nexus 9508 switch with an 9732C-EX line card support this routing mode.

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

Note For LPM heavy routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the [**no**] **system routing template-lpm-heavy** command to put the device in LPM heavy routing mode to support a larger LPM scale.

Example:

```
switch(config)# system routing template-lpm-heavy
```

Step 3 (Optional) Use the **show system routing mode** command to see the LPM routing mode.

Example:

```
switch(config)# show system routing mode
                        Configured System Routing Mode: LPM Heavy
                        Applied System Routing Mode: LPM Heavy
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure L3 host heavy template

Beginning with Cisco NX-OS Release 10.6(2)F, you can use **template-l3-host-heavy** to handle /128 host routes and increased IPv6 route scale up to the verified limit. This new template is introduced on Cisco Nexus N9300-FX2 platform switches.

Beginning with Cisco NX-OS Release 10.6(3)F, you can use this template on Cisco Nexus N9K-C9364C-H1, N9K-C9332D-H2R, and N9K-C93400LD-H1 platform switches.

TCAM Tcam space carving does not come from the template changes. To achieve the required LPM in TCAM, apply the these configurations:

- **hardware profile multicast max-limit lpm-entries 0**
- **hardware profile ipv6 lpm-entries maximum 6144**

Procedure

Step 1 Use the configure terminal. command to enter global configuration mode.

Example:

```
switch# configure terminal switch(config)#
```

Step 2 Use the [no] system routing template-l3-host-heavy command to configure L3 Host Heavy Template.

Example:

```
switch(config)# system routing template-l3-host-heavy
```

Configure LPM internet-peering routing mode (Cisco Nexus 9500-R platform switches, Cisco Nexus 9300-EX platform switches and Cisco Nexus 9000 series switches with 9700-EX line cards only)

Beginning with Cisco NX-OS Release 7.0(3)I6(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). Only the Cisco Nexus 9300-EX platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards support this routing mode.

Beginning with Cisco NX-OS Release 9.3(1), Cisco Nexus 9500-R platform switches support this routing mode.

When **system routing template-internet-peering** is configured, multicast traffic is not supported. However, it is still possible to have feature PIM and PIM configurations in the switch.

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

Note For LPM Internet-peering routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide. Cisco Nexus 9500-R platform switches in LPM Internet-peering mode scale out prectably only if they use internet-peering prefixes. If a Cisco Nexus 9500-R platform switch uses other prefix patterns, it might not achieve documented scalability numbers.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the [**no**] **system routing template-internet-peering** command to put the device in LPM Internet-peering routing mode to support IPv4 and IPv6 LPM Internet route entries.

Example:

```
switch(config)# system routing template-internet-peering
```

Step 3 (Optional) use the **show system routing mode** command to see see the LPM routing mode.

Example:

```
switch(config)# show system routing mode
                        Configured System Routing Mode: Internet Peering
                        Applied System Routing Mode: Internet Peering
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 Use the **reload** command go reboot the entire device.

Example:

```
switch(config)# reload
```

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.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 (Optional) Use the **show routing ipv4 memory estimate routes routes next-hops hops** command to see shared memory estimates to help you determine the memory requirements for routes.

Example:

```
switch(config)# show routing ipv4 memory estimate routes 262144 next-hops 32
                        Shared memory estimates:
                        Current max 512 MB; 78438 routes with 64 nhs
                        in-use 2 MB; 2642 routes with 1 nhs (average)
                        Configured max 512 MB; 78438 routes with 64 nhs
                        Estimate memory with fixed overhead: 1007 MB; 262144 routes with 32 nhs
                        Estimate with variable overhead included:
                        - With MVPN enabled VRF: 1136 MB
                        - With OSPF route (PE-CE protocol): 1375 MB
                        - With EIGRP route (PE-CE protocol): 1651 M
```

Step 3 Use the **vdc switch id id** command to specify the VDC switch ID.

Example:

```
switch(config)# vdc switch id 1
switch(config-vdc)#
```

Step 4 Use the **limit-resource u4route-mem minimum min-limit maximum max-limit** command to configure the limits for IPv4 memory in megabytes.

Example:

```
switch(config-vdc)# limit-resource u4route-mem minimum 1024 maximum 1024
```

Note Beginning with Cisco Nexus Release 10.2(2)F, this command is only applicable to the 32-bit version of the software.

Step 5 Use the **exit** command to exit the VDC configuration mode.

Example:

```
switch(config-vdc)# exit
switch(config)#
```

Step 6 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Step 7 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure LPM dual-host routing mode (Cisco Nexus 9200 and 9300-EX platform switches)

You can configure LPM heavy routing mode in order to support more LPM route entries. Only the Cisco Nexus 9200 and 9300-EX platform switches and the Cisco Nexus 9508 switch with a 9732C-EX line card support this routing mode.

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

Note For LPM heavy routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the [**no**] **system routing template-lpm-heavy** command to put the device in LPM heavy routing mode to support a larger LPM scale.

Example:

```
switch(config)# system routing template-lpm-heavy
```

Step 3 (Optional) Use the **show system routing mode** command to see the LPM routing mode.

Example:

```
switch(config)# show system routing mode

                        Configured System Routing Mode: LPM Heavy

                        Applied System Routing Mode: LPM Heavy
```

Step 4 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 Use the **reload** command to reboot the entire device.

Example:

```
switch(config)# reload
```

Configure IPv6 redirect syslog

To enable/disable the IPv6 redirect syslog or change the logging interval, use the below CLIs:

Note By default, redirecting syslog will be enabled.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Use the **ipv6 redirect syslog** [<*value*>] command to configure the syslog for excessive IPv6 redirect messages.

Example:

```
switch(config)# ip redirect syslog 60
                        switch(config)#
```

- **ipv6 redirect syslog**: Enables the syslog for IPv6 redirect messages.
- *value*: Configures the logging interval. The range is minimum 30 seconds to maximum 1800 seconds. The default interval is 60 seconds.

Step 3 (Optional) Use the **no ipv6 redirect syslog** command to disable the syslog for excessive IPv6 redirect messages.

Example:

```
switch(config)# no ipv6 redirect syslog
```

Verify the IPv6 configuration

To display the IPv6 configuration, perform one of these 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 system routing mode	Displays the LPM routing mode.
show ipv6 icmp	Displays ICMPv6 information.
show ipv6 nd	Displays IPv6 neighbor discovery interface information.
show ipv6 neighbor	Displays IPv6 neighbor entry.
show ipv6 nd addr-registry	Displays the IPv6 address registry entries of the compute node.

Verify the IPv6 configuration

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

Command	Purpose
show hardware forwarding ip verify	Displays the IPv4 and IPv6 packet verification configuration.

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.
show ipv6 icmp	Displays ICMPv6 information.
show ipv6 nd	Displays IPv6 neighbor discovery interface information.
show ipv6 neighbor	Displays IPv6 neighbor entry.
show ipv6 nd addr-registry	Displays the IPv6 address registry entries of the compute node.