

IPv4 Addresses

This chapter describes how to configure Internet Protocol version 4 (IPv4), which includes addressing, Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP), on the Cisco NX-OS device.

This chapter includes the following sections:

- IPv4 Addresses, on page 1
- Virtualization support for IPv4, on page 9
- Prerequisites for IPv4, on page 9
- Guidelines and Limitations for IPv4, on page 9
- Default settings, on page 12
- Configure IPv4, on page 13
- Verify the IPv4 Configuration, on page 32
- Additional References, on page 33

IPv4 Addresses

An IP address is a unique identifier assigned to each network interface on a device, allowing communication with other hosts across a network. Configuring IP addresses on network interfaces enables those interfaces to send and receive packets within the network.

An interface can have one primary IP address, which is the main source and destination for packets generated by the device. You can also assign multiple secondary IP addresses to an interface to support multiple subnets or network requirements. All devices on the same interface must share the same primary IP address because outgoing packets always use this primary address as their source. Every IPv4 packet includes source and destination information taken from the configured IP addresses. For more information, see Multiple IPv4 Addresses.

A subnet mask is a 32-bit value that divides an IP address into the network and host portions. By applying a subnet mask, you specify the network to which an IP address belongs and isolate the host component. Subnet masks help IP networks distinguish between network IDs and host IDs, facilitating effective routing and addressing.

The IP feature on the device manages IPv4 packet handling and forwarding, including unicast and multicast route lookup, software-based access control list (ACL) forwarding, and duplicate address detection. It also oversees IP address configuration for network interfaces, static route management, and packet transmission and reception by IP clients.

Note Nexus switches drop packets that are sent to a null0 interface. If an IPv4 or IPv6 packet is sent to a null0 interface, Cisco Nexus 9000 switches do not respond with an ICMP or ICMPv6.

Multiple IPv4 Addresses

Cisco NX-OS allows you to configure multiple IP addresses for each interface. You can set any number of secondary addresses to meet different requirements. Common scenarios include:

- When you require more host IP addresses than a single subnet allows. For instance, if your subnetting scheme supports up to 254 hosts per logical subnet but you need 300 hosts on a single physical subnet, you can apply secondary IP addresses on the routers or access servers. This enables you to map two logical subnets onto one physical subnet.
- When two subnets of the same network are physically separated by another network, you can use a secondary address to unify them. In these cases, one network is effectively extended over another. Note that a subnet must not appear on more than one active interface on the router at any time.

Note If any device on a network segment uses a secondary IPv4 address, all other devices on that same interface must use a secondary address from the same network or subnet. Inconsistent application of secondary addresses within the segment can quickly result in routing loops.

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 1: 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 2: 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 3: 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 4: 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 routing mode	Broadcom T2 mode	CLI command
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 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 5: 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), you can store host routes on the longest prefix match (LPM) table to achieve a larger host scale.

In a LPM mode, the switch supports fewer host routes. If you add more host routes than the supported scale, the excess routes from the host table occupy space in the LPM table, which reduces the total number of LPM routes available. This feature is supported on Cisco Nexus 9300 and 9500 platform switches.

In the default system routing mode, Cisco Nexus 9300 platform switches are set for higher host scale and fewer LPM routes, allowing the LPM space to be used for additional host routes. On Cisco Nexus 9500 platform switches, this feature is supported on line cards only in the default system routing and nonhierarchical routing modes. Fabric modules do not support this feature.

Address Resolution Protocol

Networking devices and Layer 3 switches use Address Resolution Protocol (ARP) to map IP addresses (network layer) to Media Access Control (MAC) address (physical layer) to enable IP packets to be sent across networks.

Before a device sends a packet to another device, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network.

Each device compares the IP address to its own. Only the device with the matching IP address replies to the device that sends the data with a packet that contains the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. The following figure shows the ARP broadcast and response process.

Figure 1: ARP Process



When the destination device lies on a remote network that is beyond another device, the process is the same except that the device that sends the data sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet. ARP is enabled by default.

The default system-defined CoPP policy rate limits ARP broadcast packets bound for the supervisor module. The default system-defined CoPP policy prevents an ARP broadcast storm from affecting the control plane traffic but does not affect bridged packets.

ARP cache

The Address Resolution Protocol (ARP) cache is a table maintained by network devices such as routers and switches to store mappings between IP addresses and their corresponding MAC hardware addresses.

ARP cache minimizes broadcasts and limits wasteful use of network resources. The mapping of IP addresses to MAC addresses occurs at each hop (device) on the network for every packet sent over an internetwork, which can affect network performance.

- ARP caching stores network addresses and the associated data-link addresses in the memory for a period of time, which minimizes the use of valuable network resources to broadcast for the same address each time that a packet is sent.
- You must maintain the cache entries that are set to expire periodically because the information might become outdated.

- Every device on a network updates its tables as addresses are broadcast.

Static and dynamic entries in the ARP Cache

Static route

Static routing requires you to configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each device.

Static routing requires more work to maintain the route table. You must update the table each time you add or change routes.

Dynamic route

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other.

Dynamic routing is more efficient than static routing because the route table is automatically updated unless you add a time limit to the cache. The default time limit is 25 minutes, but you can modify the time limit if the network has many routes that are added and deleted from the cache.

Devices that do not use ARP

When a network is segmented into two parts, a bridge connects these segments and filters traffic based solely on MAC addresses. The bridge maintains its own MAC address table to manage this filtering. In contrast, a device maintains an ARP cache that maps both IP addresses and their corresponding MAC addresses.

Passive hubs serve as central connection points that physically link devices within a network. Operating at Layer 1 (the physical layer), they broadcast incoming messages to all ports without maintaining any address table.

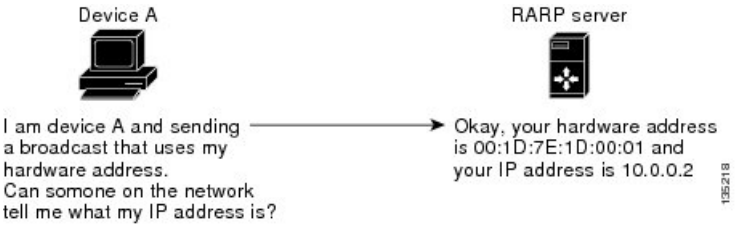
Layer 2 switches determine which port of a device receives a message that is sent only to that port. However, Layer 3 switches are devices that build an ARP cache (table).

Reverse ARP

Reverse ARP (RARP) works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. RARP is defined by RFC903.

RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned into the hardware. Use of RARP requires an RARP server on the same network segment as the router interface. The Figure 2: Reverse ARP, on page 7 figure shows how RARP works.

Figure 2: Reverse ARP



RARP has several limitations. Because of these limitations, most businesses use Dynamic Host Configuration Protocol (DHCP) to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP.

These are the most important limitations:

- Because RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts and not subnet masks or default gateways.

Proxy ARP

Proxy ARP is a feature that allows a device, typically a router, to answer ARP requests on behalf of another device. This allows a host to communicate with devices on remote subnets without needing to configure routing or a default gateway.

These are the advantages of proxy ARP:

- Proxy ARP enables a device that is physically located on one network to be logically part of a different physical network connected to the same device or firewall.
- Proxy ARP allows you to hide a device with a public IP address on a private network behind a router, yet still have the device appear to be on the public network in front of the router. By hiding its identity, the router accepts responsibility for routing packets to the real destination.
- Proxy ARP can help devices on a subnet reach remote subnets without configuring routing or a default gateway.

When devices are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the devices does not send a broadcast message because routers do not pass hardware-layer broadcasts and the addresses cannot be resolved.

When you enable proxy ARP on the device and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The device responds as if it is the remote destination for which the broadcast is addressed, with an ARP response that associates the device’s MAC address with the remote destination's IP address. The local device believes that it is directly connected to the destination, while in reality its packets are being forwarded from the local subnetwork toward the destination subnetwork by their local device. By default, proxy ARP is disabled.

Local proxy ARP

Local proxy ARP enables a device to respond to ARP requests for IP addresses within a subnet where normally no routing is required.

When you enable local proxy ARP, ARP responds to all ARP requests for IP addresses within the subnet and forwards all traffic between hosts in the subnet. Use this feature only on subnets where hosts are intentionally prevented from communicating directly by the configuration on the device to which they are connected.

Gratuitous ARP

Gratuitous ARP sends a request with an identical source IP address and a destination IP address to detect duplicate IP addresses.

Cisco NX-OS supports enabling or disabling gratuitous ARP requests or ARP cache updates.

Periodic ARP refresh on MAC delete

The ARP process tracks the MAC deletes and sends the periodic ARP Refresh on the L3 VLAN interface in a configured interval of time for the configured count. If the MAC is learned, ARP process stops sending the periodic ARP Refreshes.

For more information, see Configuring periodic ARP refresh on MAC delete for SVIs, on page 25.

ARP refresh on MAC Delete

MAC addresses are deleted or flushed when triggers such as interface flaps, clearing the MAC address table, or STP TCN notifications occur. Once the MAC addresses are deleted, the ARP process initiates ARP Refresh for all adjacencies associated with these MAC addresses. These ARP Refreshes are subject to CoPP limits, which can prevent the ARP process from effectively relearning all ARP entries.

Beginning with Cisco NX-OS Release 10.5(3)F, instead of sending Refresh retries at 2, 4, 8 seconds and deleting after 16 seconds (if no response is received), the ARP process sends the Refreshes at 2, 4, 8, and 8 seconds, and deletes after an additional 8 seconds (one extra Refresh). This change improves ARP learning with scaled ARP entries, even if some ARP response packets are subject to CoPP limits.

Glean throttles

Glean throttling is a mechanism where, if an Address Resolution Protocol (ARP) request for the next-hop IP address is not resolved by a line card when forwarding incoming IP packets, the line card forwards those packets to the supervisor for resolution. The supervisor resolves the MAC address for the next hop and programs the hardware.

When an ARP request is sent, the software adds a /32 drop adjacency in the hardware to prevent the packets to the same next-hop IP address to be forwarded to the supervisor. When the ARP is resolved, the hardware entry is updated with the correct MAC address. If the ARP entry is not resolved before a timeout period, the entry is removed from the hardware.



Note Glean throttling is supported for IPv4 and IPv6, but IPv6 link-local addresses are not supported.

Path MTU discovery

Path maximum transmission unit (MTU) discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection.

It is described in RFC 1191. Existing connections are not affected when this feature is turned on or off.

ICMP

You can use the Internet Control Message Protocol (ICMP) to provide message packets that report errors and other information that is relevant to IP processing.

ICMP generates error messages, such as ICMP destination unreachable messages, ICMP Echo Requests (which send a packet on a round trip between two hosts) and Echo Reply messages. ICMP also provides many diagnostic functions and it can send and redirect error packets to the host. By default, ICMP is enabled.

Some of the ICMP message types are as follows:

- Network error messages
- Network congestion messages
- Troubleshooting information
- Timeout announcements

Note

- ICMP redirects are disabled on interfaces where the local proxy ARP feature is enabled.
- The ICMP redirect feature is not supported on the Cisco N93-C64E-SG2-Q switch platform. If ICMP redirect is currently enabled on the Cisco N93-C64E-SG2-Q switch, it must be disabled.

Virtualization support for IPv4

IPv4 supports virtual routing and forwarding (VRF) instances.

Prerequisites for IPv4

IPv4 has the following prerequisites:

- IPv4 can only be configured on Layer 3 interfaces.

Guidelines and Limitations for IPv4

IPv4 has the following 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.
- You can configure a secondary IP address only after you configure the primary IP address.
- Local proxy ARP is not supported for an interface with more than one HSRP group that belongs to multiple subnets.
- The **ip proxy-arp** command is not supported in a VXLAN EVPN Fabric specifically on an SVI which is enabled with **fabric forwarding mode anycast-gateway** .
- Local proxy ARP does not work on the VLAN when you disable ARP suppression on the VLAN.
- For Cisco Nexus 9500 platform switches with -R line cards, 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.
- 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.
- 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 IPv4 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 following switches and line cards:
  - Cisco Nexus 9804 switches.
  - Cisco Nexus X98900CD-A, and X9836DM-A line cards with 9804, and 9808 platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, dynamic routing is supported on the Cisco Nexus 9808 platform 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(2)F, static routing is supported on Cisco Nexus 9232E-B1 switch.
- Beginning with Cisco NX-OS Release 10.2(4)M, periodic ARP Refresh on MAC delete support is provided on Cisco Nexus 9000 Series platform switches with the following limitations:
  - During configuration of the **ip arp refresh-adj-on-mac-delete retry** command, ARP process does not trigger Refresh although ARP is learned and MAC is not learned. It tries to send periodic ARP Refreshes on MAC delete/flush.
  - The periodic ARP Refresh behavior is triggered for the MAC's deletion after configuring the **ip arp refresh-adj-on-mac-delete retry** command.

- The trigger for this periodic ARP Refresh is MAC delete. This feature does not address the MAC learn miss on receiving burst packets.
- During configuring, you must choose the right count and interval based on the scale/network requirements.

- Beginning with Cisco NX-OS Release 10.4(1)F, out of subnet ARP resolution support is provided on Cisco Nexus 9000 Series platform switches for the following L3 interfaces:
  - Ethernet
  - Sub-interfaces
  - Port-channel
  - FEX
  - IP unnumbered interface



Note
- The out of subnet ARP resolution feature is not supported on SVI L3 interfaces and on VPC or HSRP or VXLAN deployments.

- Beginning with Cisco NX-OS Release 10.4(2)F, the **ip arp cache intf-limit** configuration is supported to limit the ARP cache entries per interface on Cisco NX-OS devices with the following capabilities:
  - Supported on global and interface modes. However, interface mode configuration takes the precedence over global mode.
  - Supported only on the following L3 interfaces:
    - SVI
    - SVI Unnumbered Interfaces
  - Not supported on the following L3 interfaces:
    - Ethernet
    - Subinterfaces
    - Port-channel
    - Unnumbered interfaces
  - If the configuration is applied to non-supporting interfaces, this configuration will be applied to the global mode.

- Beginning with Cisco NX-OS Release 10.4(2)F, the following IPv4 Layer-3 features are supported on the Cisco Nexus 9232E-B1 switches:
  - ARP
  - ECMP
  - IPv4 unicast routes

    - IPv4 host routes

- Beginning with Cisco NX-OS Release 10.4(2)F, you can configure all NX-OS features using Class E IP address. If the IPv4 address space is exhausted, you can configure the features using the Class E IP address. The following switches and line cards are supported for Class E IP address:
    - Cisco Nexus 92348GC-X
    - Cisco Nexus 9300-EX/FX/FX2/FX3/H2R/H1/GX/GX2
    - Cisco Nexus 9300C
    - Cisco Nexus 9700-EX/FX/GX line cards
- Beginning with Cisco NX-OS Release 10.6(1)F, the following IPv4 Layer-3 features are supported on the Cisco Nexus N9336C-SE1 switches:
    - IPv4 unicast routes
    - IPv4 host routes
    - IPv4 Neighbor Discovery
    - ARP
    - IP directed broadcast
    - Non SVI IP unnumbered interface
- Beginning with Cisco NX-OS Release 10.6(2)F, you can configure hardware forwarding of IP directed broadcasts using new **hw-assist** keyword in IP directed broadcast configuration.
    - This feature is supported only on L3 topology
    - This feature is not supported on VXLAN
    - Keyword hw-assist is supported only on SVIs
    - This feature is supported on Cisco Nexus 9300-FX/FX2/FX3/GX/GX2/HX based TORs. It is not supported for EOR.
- Beginning with Cisco NX-OS Release 10.6(2)F, following features are supported on Cisco Nexus N9324C-SE1U and N9348Y2C6D-SE1U platform switches
    - Non SVI IP unnumbered interface
    - IP directed broadcast
- Beginning with Cisco NX-OS Release 10.6(2n), Layer 3 support for IPv4 is supported on the Cisco N9164E-NS4-O switch.

Default settings

The table below lists the default settings for IP parameters.

Parameters	Default
ARP timeout	1500 seconds
Proxy ARP	Disabled

Configure IPv4



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

Configure IPv4 address

You can assign a primary IP address for a network interface.

Procedure

Step 1 Use the **configure terminal** command to 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 **ip address** *ip-address/length* [*secondary*] command to specify a primary or secondary IPv4 address for an interface.

Example:

```
switch(config-if)# ip address
192.2.1.1 255.0.0.0
```

- The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address.
- The network mask can be indicated as a slash (/) and a number, which is the prefix length. 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). A slash must precede the decimal value and there must be no space between the IP address and the slash.

Step 4 (Optional) Use the **show ip interface** command to display interfaces configured for IPv4.

Example:

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

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

Example:

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

Configure multiple IP address

You can only add secondary IP addresses after you configure primary IP addresses.

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 **ip address** *ip-address/length* [*secondary*] command to specify a the configured address as a secondary IPv4 address.

Example:

```
switch(config-if)# ip address
192.168.1.1 255.0.0.0 secondary
```

Step 4 (Optional) Use the **show ip interface** command to see interfaces configured for IPv4.

Example:

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

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

Example:

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

Configure max-host routing mode

By default, Cisco NX-OS 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, refer to 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 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 root the entire device.

Example:

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

Configure nonhierarchical routing mode (Cisco Nexus 9500 platform 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**] command to put the line cards in Broadcom T2 mode 3.

Example:

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

Alternatively, Broadcom T2 mode 4 if you use the **max-l3-mode** option to support a larger LPM scale. 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 saves 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 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 more LPM route entries. Only the Cisco Nexus 9200 and 9300-EX platform 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
```

Displays the LPM routing mode.

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
```

Configuring LPM Internet-Peering Routing Mode

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).

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

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

Note For LPM Internet-peering routing mode scale numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide. Cisco Nexus 9500-R platform switches in LPM Internet-peering mode scale out predictably only if they use internet-peering prefixes. If Cisco Nexus 9500-R platform switches use 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 the LPM 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 to reboot the entire device.

Example:

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

Configure LPM dual-host routing mode

Beginning with Cisco NX-OS Release 7.0(3)I5(1), you can configure LPM dual-host routing mode in order to increase the ARP/ND scale to double the default mode value. Only the Cisco Nexus 9200 and 9300-EX platform switches support this routing mode.

Beginning with Cisco NX-OS Release 10.3(1)F, the **system routing template-dual-stack-host-scale** profile supports Multicast and VXLAN on Cisco Nexus 9300-FX3/GX/GX2B ToR switches, and Nexus 9408 switches.

Note Ensure that the **system routing template-dual-stack-host-scale** profile is not used with BGW.

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

Note For LPM dual-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 template-dual-stack-host-scale** command to put the device in LPM dual-host routing mode to support a larger ARP/ND scale.

Example:

```
switch(config)# system routing template-dual-stack-host-scale
                       Warning: The command will take effect after next reload.
                       Note: This requires copy running-config to startup-config before switch
reload.
```

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

Example:

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

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 a Static ARP Entry

You can configure a static ARP entry on the device to map IP addresses to MAC hardware addresses, including static multicast MAC addresses.

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 **ip arp address** *ip-address mac-address* command to associate an IP address with a MAC address as a static entry.

Example:

```
switch(config-if)# ip arp 192.168.1.1
0019.076c.1a78
```

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

Example:

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

Configure proxy ARP

Configure proxy ARP on the device to determine the media addresses of hosts on other networks or subnets.

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 **ip proxy arp** command to enable proxy ARP on the interface.

Example:

```
switch(config-if)# ip proxy arp
```

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

Example:

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

Configure local proxy ARP on ethernet interfaces

You can configure local proxy ARP on Ethernet interfaces.

Before you begin

Local proxy ARP is supported only on Layer 3 Ethernet interfaces. It is not supported on Layer 2 Ethernet interfaces.

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 [**no**]**ip local-proxy-arp** command to enable Local Proxy ARP on the interface.

Example:

```
switch(config-if)# ip local-proxy-arp
```

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

Example:

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

Configuring local proxy ARP on SVIs

You can configure local proxy ARP on SVIs, and beginning with Cisco NX-OS Release 7.0(3)I7(1), you can suppress ARP broadcasts on corresponding VLANs.

Before you begin

If you are planning to suppress ARP broadcasts, configure the double-wide ACL TCAM region size for ARP/Layer 2 Ethertype using the hardware access-list tcam region arp-ether 256 double-wide command, save the configuration, and reload the switch. (For more information, see the Configuring ACL TCAM Region Sizes section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.)

Procedure

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

Example:

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

Step 2 Use the **interface vlan vlan-id** command to create a VLAN interface and enters the configuration mode for the SVI.

Example:

```
switch(config)# interface vlan 5
switch(config-if)#
```

Step 3 Use the [**no**] **ip local-proxy-arp** [**reachable**] command to enable local proxy ARP on SVIs.

Example:

```
switch(config-if)# ip local-proxy-arp reachable
```

Use the **reachable** option to enables enhanced proxy ARP which ensures ARP responses are sent only if the destination host is known or reachable. If unknown, a glean ARP request is generated to discover the destination.

Note Beginning with Cisco NX-OS Release 10.6(3)F, the **no-hw-flooding** option is deprecated and its functionality is now included as part of the default **ip local-proxy-arp** command.

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

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

Configuring periodic ARP refresh on MAC delete for SVIs

Beginning with Cisco NX-OS Release 10.2(4)M, you can configure periodic ARP Refresh on MAC delete for SVIs.

By default this command is disabled. This command must be configured under the SVI for the periodic ARP Refreshes to learn the MACs from the ARP response packet for the silent hosts on MAC delete/flush.

Procedure

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

Example:

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

Step 2 Use the **interface vlan vlan-id** command to create a VLAN interface and enter the configuration mode for the SVI.

Example:

```
switch(config)# interface vlan 5
                        switch(config-if)#
```

Step 3 Use the [**no**] **ip arp refresh-adj-on-mac-delete retry** [**count** <*frequency count*>] [**interval** <*interval in sec*>] command to configure the ARP Refreshes to learn the MACs from the ARP response packet for the silent hosts on MAC delete/flush.

Example:

```
switch(config-if)# ip arp refresh-adj-on-mac-delete retry count 3 interval 15
                        switch(config-if)#
```

- <*frequency count*>: The range is 1–3. The default is 3.
- <*interval in sec*>: The range is 1–60 seconds. The default is 15 seconds.

Note If the interval is greater than 3/4th of ARP Refresh time, this command is rejected with the below message:

```
ARP refresh will be sent earlier to the interval due to ARP timeout
configuration. This configuration is not useful.
```

Step 4 (Optional) Use the **copy running-config startup-config** command to copy the running configuration to the startup configuration.

Example:

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

Configure gratuitous ARP

You can configure gratuitous ARP on an 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 **ip arp gratuitous** {**request** | **update**] command to enable gratuitous ARP on the interface.

Example:

```
switch(config-if)# ip arp gratuitous
request
```

Gratuitous ARP is enabled by default.

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

Example:

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

Configure out of subnet ARP resolution

Beginning with Cisco NX-OS Release 10.4(1)F, you can enable or disable out of subnet ARP resolution using the **ip arp outside-subnet** command.

This command is available on both global and interface mode. There is no impact on config-replace and dual stage commit when this command is enabled.



Note When this command is enabled, downgrade from Cisco NX-OS Release 10.4(1)F is restricted, and user will be prompted with an error message to remove the out of subnet ARP resolution configuration before proceeding for downgrade.

Procedure

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

Example:

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

Step 2 Use the **[no] ip arp outside-subnet** command to enable or disables the ARP out of subnet packet transaction on connected host.

Example:

```
switch(config)# ip arp outside-subnet
```

Step 3 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

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

Configure ARP cache limit per SVI interface

Beginning from Cisco NX-OS Release 10.4(2)F, you can set the number of maximum ARP cache entries to be allowed per SVI interface on the Cisco NX-OS devices. This configuration is supported on both global and interface modes.

Procedure

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

Example:

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

Step 2 Use the **interface vlan** *vlan-id* command to create a VLAN interface and enters the configuration mode for the SVI.

Example:

```
switch(config)# interface vlan 5
 switch(config-if)#
```

Step 3 Use the **[no] ip arp cache intf-limit** *value* command to configure the set limit of ARP cache entries for the SVI interface

Example:

```
switch(config-if)# ip arp cache intf-limit 50000
 switch(config-if)#
```

Range of valid ARP entries is 1-128000.

intf-limit: Specifies the number of valid dynamic ARP entries per interface.

The **no** form of this command removes the configuration.

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

Example:

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

Configure path MTU discovery

You can configure path MTU discovery.

Procedure

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

Example:

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

Step 2 Use the **ip tcp path-mtu-discovery** command to enable path MTU discovery.

Example:

```
switch(config)# ip tcp
path-mtu-discovery
```

Step 3 (Optional) Use the **copy running-config startup-config** commad to save this configuration change.

Example:

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

Configure IP directed broadcasts

IP directed broadcast feature forwards a packet from a node, which is not part of the destination subnet, at the link-layer in the destination subnet post IP routing

A device that is not directly connected to its destination subnet forwards an IP directed broadcast in the same way it forwards unicast IP packets destined to a host on that subnet. When a directed broadcast packet reaches a device that is directly connected to its destination subnet, that packet is broadcast on the destination subnet.

The destination MAC address is rewritten as broadcast address and the packet is forwarded as a link-layer broadcast.

If directed broadcast is enabled for an interface, incoming IP packets whose addresses identify them as directed broadcasts intended for the subnet to which that interface is attached are broadcasted on that subnet. You can optionally filter those broadcasts through an IP access list such that only those packets that pass through the access list are broadcasted on the subnet.

Beginning with Cisco NX-OS Release 10.6(2)F, new keyword hw-assist is introduced that enables hardware forwarding of IP directed broadcasts.

To enable IP directed broadcasts, use the following command in the interface configuration mode:

Procedure

Use the **ip directed-broadcast**[*acl* | **hw-assist**] command to enable the translation of a directed broadcast to physical broadcasts.

Example:

```
switch(config-if) # ip directed-broadcast
```

You can optionally filter those broadcasts through an IP access list.

CLI **ip directed-broadcast** enables software forwarding of IP directed broadcasts.

- acl: filters the IP directed broadcast packets through the specified IP access list.
- hw-assist: enables hardware forwarding of IP directed broadcasts.

You can either use the **ip directed-broadcast** *acl* command or the **ip directed-broadcast hw-assist** command on an interface. If you configure both the commands, the latest configuration takes precedence.

Configuring IP glean throttling

We recommend that you configure IP glean throttling to filter the unnecessary glean packets that are sent to the supervisor for ARP resolution for the next hops that are not reachable or do not exist. IP glean throttling boosts software performance and helps to manage traffic more efficiently.

Note Glean throttling is supported for IPv4 and IPv6, but IPv6 link-local addresses are not supported.

SUMMARY STEPS

1. Use the **configure terminal** command to enter global configuration mode.
2. Use the [**no**] **hardware ip glean throttle** command to enable IP glean throttling.
3. (Optional) Use the **copy running-config startup-config** command to save this configuration change.

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	Use the **configure terminal** command to enter global configuration mode.
	Example:
	`switch# configure terminal`
	`switch(config)#`	

	Command or Action	Purpose
Step 2	Use the [**no**] **hardware ip glean throttle** command to enable IP glean throttling. **Example:** `switch(config) # hardware ip glean throttle`	
Step 3	(Optional) Use the **copy running-config startup-config** command to save this configuration change. **Example:** `switch(config)# copy running-config startup-config`	

Configure the hardware IP glean throttle maximum

You can limit the maximum number of drop adjacencies that are installed in the Forwarding Information Base (FIB).

Procedure

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

Example:

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

Step 2 Use the [**no**] **hardware ip glean throttle maximum** *count* command to configure the number of drop adjacencies that are installed in the FIB.

Example:

```
switch(config) # hardware ip glean
throttle maximum 2134
```

Step 3 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

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

Configure the hardware IP glean throttle timeout

You can configure a timeout for the installed drop adjacencies to remain in the FIB.

SUMMARY STEPS

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

2. Use the [**no**] **hardware ip glean throttle maximum timeout** *timeout-in-seconds* command to configure the timeout for the installed drop adjacencies to remain in the FIB.
3. (Optional) Use the **copy running-config startup-config** command save this configuration change.

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	Use the **configure terminal** command to enter global configuration mode.
	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	Use the [**no**] **hardware ip glean throttle maximum timeout** *timeout-in-seconds* command to configure the timeout for the installed drop adjacencies to remain in the FIB. **Example:** `switch(config)# hardware ip glean throttle maximum timeout 300`	The range is from 300 seconds (5 minutes) to 1800 seconds (30 minutes). **Note** After the timeout period is exceeded, the
Step 3	(Optional) Use the **copy running-config startup-config** command save this configuration change. **Example:** `switch(config)# copy running-config startup-config`	

Configure the interface IP address for the ICMP source IP field

You can configure an interface IP address for the ICMP source IP field to handle ICMP error messages.

Procedure

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

Example:

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

Step 2 Use the [**no**] **ip source** {**ethernet** *slot/port* | **loopback** *number* | **port-channel** *number*} **icmp-errors** command tp configure an interface IP address for the ICMP source IP field to route ICMP error messages.

Example:

```
switch(config)# ip source loopback 0
icmp-errors
```

Step 3 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

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

Configure IPv4 redirect syslog

To enable/disable the IPv4 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 **ip redirect syslog** [<*value*>] command to configure the syslog for excessive IP redirect messages.

Example:

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

- **ip redirect syslog**: Enables the syslog for IPv4 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 ip redirect syslog** command to disable the syslog for excessive IPv4 redirect messages.

Example:

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

Verify the IPv4 Configuration

To display the IPv4 configuration information, perform one of the these tasks:

Command	Purpose
show ip adjacency	Displays the adjacency table.

Command	Purpose
show ip adjacency summary	Displays the summary of number of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays the summary of the number of throttle adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [**vrf** *vrf-name*]	Displays the ARP statistics.
show ip arp internal info interface <*interface-name*>	Displays the configured count and interval

Additional References

Related Documents for IPv4

Related Topic	Document Title
TCAM regions	See the Configuring ACL TCAM Region Sizes section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.