



Configuring IPv4

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:

- [About IPv4, page 2-1](#)
- [Licensing Requirements for IPv4, page 2-7](#)
- [Prerequisites for IPv4, page 2-7](#)
- [Guidelines and Limitations for IPv4, page 2-7](#)
- [Default Settings, page 2-7](#)
- [Configuring IPv4, page 2-7](#)
- [Verifying the IPv4 Configuration, page 2-20](#)

About IPv4

You can configure IP on the device to assign IP addresses to network interfaces. When you assign IP addresses, you enable the interfaces and allow communication with the hosts on those interfaces.

You can configure an IP address as primary or secondary on a device. An interface can have one primary IP address and multiple secondary addresses. All networking devices on an interface should share the same primary IP address because the packets that are generated by the device always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address. For more information, see the [“Multiple IPv4 Addresses” section on page 2-2](#).

You can use a subnet to mask the IP addresses. A mask is used to determine what subnet an IP address belongs to. An IP address contains the network address and the host address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The IP feature is responsible for handling IPv4 packets that terminate in the supervisor module, as well as forwarding of IPv4 packets, which includes IPv4 unicast/multicast route lookup and software access control list (ACL) forwarding. The IP feature also manages the network interface IP address configuration, duplicate address checks, static routes, and packet send/receive interface for IP clients.

This section includes the following topics:

- [Multiple IPv4 Addresses, page 2-2](#)
- [LPM Routing Modes, page 2-2](#)
- [Address Resolution Protocol, page 2-3](#)
- [ARP Caching, page 2-4](#)
- [Static and Dynamic Entries in the ARP Cache, page 2-4](#)
- [Devices That Do Not Use ARP, page 2-4](#)
- [Reverse ARP, page 2-4](#)
- [Proxy ARP, page 2-5](#)
- [Local Proxy ARP, page 2-6](#)
- [Gratuitous ARP, page 2-6](#)
- [Glean Throttling, page 2-6](#)
- [Path MTU Discovery, page 2-6](#)
- [ICMP, page 2-6](#)
- [Virtualization Support, page 2-7](#)

Multiple IPv4 Addresses

Cisco NX-OS supports multiple IP addresses per interface. You can specify an unlimited number of secondary addresses for a variety of situations. The most common are as follows:

- When there are not enough host IP addresses for a particular network interface. For example, if your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you must have 300 host addresses, then you can use secondary IP addresses on the routers or access servers to allow you to have two logical subnets that use one physical subnet.
- Two subnets of a single network might otherwise be separated by another network. You can create a single network from subnets that are physically separated by another network by using a secondary address. In these instances, the first network is extended, or layered on top of the second network. A subnet cannot appear on more than one active interface of the router at a time.



Note

If any device on a network segment uses a secondary IPv4 address, all other devices on that same network interface must also use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause 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 significantly more LPM route entries.

The following tables list the LPM routing modes that are supported on the Cisco Nexus 9300 Series and 9500 Series switches.

Table 2-1 LPM Routing Modes for Cisco Nexus 9300 Series Switches

LPM Routing Mode	Broadcom T2 Mode	CLI Command
Default system routing mode	3	

Table 2-2 LPM Routing Modes for Cisco Nexus 9500 Series Switches

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

For detailed configuration information, see the “Configuring IPv4” section on page 2-7.

Address Resolution Protocol

Networking devices and Layer 3 switches use Address Resolution Protocol (ARP) to map IP (network layer) addresses to (Media Access Control [MAC]-layer) addresses 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. Figure 2-1 shows the ARP broadcast and response process.

Figure 2-1 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 Caching

ARP caching 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 may 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 routing requires that you manually 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 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 divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table, which uses MAC addresses only. A device has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. They send messages out on all their ports to the devices and operate at Layer 1 but do not maintain an address table.

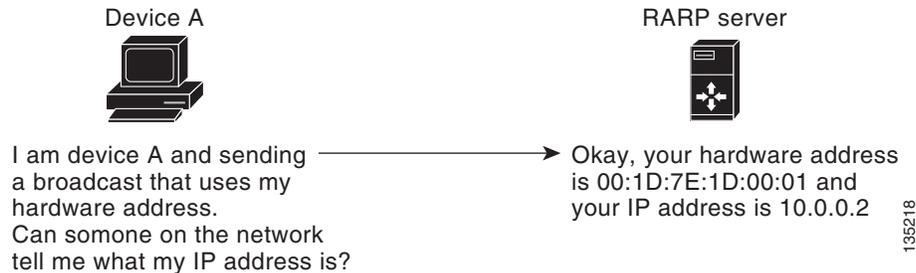
Layer 2 switches determine which port of a device receives a message that is sent only to that port. However, Layer 3 switches are devices that build an ARP cache (table).

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. 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. [Figure 2-2](#) shows how RARP works.

Figure 2-2 Reverse ARP



RARP has several limitations. Because of these limitations, most businesses use DHCP to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP. The following 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 enables a device that is physically located on one network appear 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 and 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

You can use local Proxy ARP to enable 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.

Glean Throttling

If the Address Resolution Protocol (ARP) request for the next hop is not resolved when incoming IP packets are forwarded in a line card, the line card forwards the packets to the supervisor (glean throttling). 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.

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 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.

Virtualization Support

IPv4 supports virtual routing and forwarding (VRF) instances.

Licensing Requirements for IPv4

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	IP requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

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:

- You can configure a secondary IP address only after you configure the primary IP address.

Default Settings

Table 2-3 lists the default settings for IP parameters.

Table 2-3 Default IP Parameters

Parameters	Default
ARP timeout	1500 seconds
Proxy ARP	Disabled

Configuring IPv4

This section includes the following topics:

- [Configuring IPv4 Addressing, page 2-8](#)
- [Configuring Multiple IP Addresses, page 2-9](#)
- [Configuring Max-Host Routing Mode \(Cisco Nexus 9500 Series Switches Only\), page 2-10](#)
- [Configuring Nonhierarchical Routing Mode \(Cisco Nexus 9500 Series Switches Only\), page 2-11](#)

- [Configuring 64-Bit ALPM Routing Mode \(Cisco Nexus 9500 Series Switches Only\)](#), page 2-12
- [Configuring a Static ARP Entry](#), page 2-14
- [Configuring Proxy ARP](#), page 2-14
- [Configuring Local Proxy ARP](#), page 2-15
- [Configuring Gratuitous ARP](#), page 2-16
- [Configuring Path MTU Discovery](#), page 2-17
- [Configuring IP Directed Broadcasts](#), page 2-17
- [Configuring IP Glean Throttling](#), page 2-18
- [Configuring the Hardware IP Glean Throttle Maximum](#), page 2-18
- [Configuring a Hardware IP Glean Throttle Timeout](#), page 2-19
- [Configuring the Interface IP Address for the ICMP Source IP Field](#), page 2-20

**Note**

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

Configuring IPv4 Addressing

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

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.

	Command	Purpose
Step 3	<p>ip address <i>ip-address/length</i> [secondary]</p> <p>Example: switch(config-if)# ip address 192.168.1.1 255.0.0.0</p>	<p>Specifies a primary or secondary IPv4 address for an interface.</p> <ul style="list-style-type: none"> • 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 is no space between the IP address and the slash.
Step 4	<p>show ip interface</p> <p>Example: switch(config-if)# show ip interface</p>	(Optional) Displays interfaces configured for IPv4.
Step 5	<p>copy running-config startup-config</p> <p>Example: switch(config-if)# copy running-config startup-config</p>	(Optional) Copies the running configuration to the startup configuration.

Configuring Multiple IP Addresses

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

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.168.1.1 255.0.0.0 secondary	Specifies the configured address as a secondary IPv4 address.
Step 4	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Max-Host Routing Mode (Cisco Nexus 9500 Series Switches Only)

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

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

**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 on page 2-11 to configure the device to program all of 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*.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing max-mode host**
3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing max-mode host Example: switch(config)# system routing max-mode host	Puts the line cards in Broadcom T2 mode 2 and the fabric modules in Broadcom T2 mode 3 to increase the number of supported hosts.
Step 3	show forwarding route summary Example: switch(config)# show forwarding route summary	(Optional) Displays the LPM mode.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload	Reboots the entire device.

Configuring 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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing non-hierarchical-routing [max-l3-mode]**

3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing non-hierarchical-routing [max-l3-mode] Example: switch(config)# system routing non-hierarchical-routing max-l3-mode	Puts 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. 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	show forwarding route summary 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	(Optional) Displays the LPM mode.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload	Reboots the entire device.

Configuring 64-Bit ALPM Routing Mode (Cisco Nexus 9500 Series 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 significantly more route entries. Using 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 $2x + y \leq 128,000$



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*.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing mode hierarchical 64b-alpm Example: switch(config)# system routing mode hierarchical 64b-alpm	Causes all IPv4 and IPv6 LPM routes with a mask length that is less than or equal to 64 to be programmed in the fabric module. All host routes for IPv4 and IPv6 and all LPM routes with a mask length of 65 through 127 are programmed in the line card.
Step 3	show forwarding route summary Example: switch(config)# show forwarding route summary	(Optional) Displays the LPM mode.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload	Reboots the entire device.

Configuring a Static ARP Entry

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



Note

Broadcast and multicast MAC addresses are not supported.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip arp** *ipaddr mac_addr*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip arp <i>ipaddr mac_addr</i> Example: switch(config-if)# ip arp 192.168.1.1 0019.076c.1a78	Associates an IP address with a MAC address as a static entry.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Proxy ARP

You can configure Proxy ARP on the device to determine the media addresses of hosts on other networks or subnets.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip proxy-arp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip proxy-arp Example: switch(config-if)# ip proxy-arp	Enables Proxy ARP on the interface.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Local Proxy ARP

You can configure Local Proxy ARP on the device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip local-proxy-arp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.

	Command	Purpose
Step 3	ip local-proxy-arp Example: switch(config-if)# ip local-proxy-arp	Enables Local Proxy ARP on the interface.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ip arp gratuitous {request | update}**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip arp gratuitous {request update} Example: switch(config-if)# ip arp gratuitous request	Enables gratuitous ARP on the interface. The default is enabled.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Path MTU Discovery

You can configure path MTU discovery.

SUMMARY STEPS

1. **configure terminal**
2. **ip tcp path-mtu-discovery**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip tcp path-mtu-discovery Example: switch(config)# ip tcp path-mtu-discovery	Enables path MTU discovery.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring IP Directed Broadcasts

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for some IP subnet but which originates from a node that is not itself part of that destination subnet.

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 address in the IP header of the packet is rewritten to the configured IP broadcast address for the subnet, and the packet is sent 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.

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

Command	Purpose
ip directed-broadcast [<i>acl</i>]	Enables the translation of a directed broadcast to physical broadcasts. You can optionally filter those broadcasts through an IP access list.

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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] hardware ip glean throttle**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] hardware ip glean throttle Example: switch(config)# hardware ip glean throttle	Enables IP glean throttling.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring the Hardware IP Glean Throttle Maximum

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

SUMMARY STEPS

1. **configure terminal**
2. **[no] hardware ip glean throttle maximum *count***
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<code>[no] hardware ip glean throttle maximum count</code> Example: switch(config)# hardware ip glean throttle maximum 2134	Configures the number of drop adjacencies that are installed in the FIB.
Step 3	<code>copy running-config startup-config</code> Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring a Hardware IP Glean Throttle Timeout

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

SUMMARY STEPS

1. `configure terminal`
2. `[no] hardware ip glean throttle maximum timeout timeout`
3. (Optional) `copy running-config startup-config`

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<code>[no] hardware ip glean throttle maximum timeout timeout</code> Example: switch(config)# hardware ip glean throttle maximum timeout 300	Configures the timeout for the installed drop adjacencies to remain in the FIB. The range is from 300 seconds (5 minutes) to 1800 seconds (30 minutes). Note After the timeout period is exceeded, the drop adjacencies are removed from the FIB.
Step 3	<code>copy running-config startup-config</code> Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring 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.

SUMMARY STEPS

1. **configure terminal**
2. **[no] ip source {ethernet *slot/port* | loopback *number* | port-channel *number*} icmp-errors**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] ip source {ethernet <i>slot/port</i> loopback <i>number</i> port-channel <i>number</i>} icmp-errors Example: switch(config)# ip source loopback 0 icmp-errors	Configures an interface IP address for the ICMP source IP field to route ICMP error messages.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Verifying the IPv4 Configuration

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

Command	Purpose
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays a summary of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays a summary of the number of throttle adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [vrf <i>vrf-name</i>]	Displays the ARP statistics.