



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 switch.

This chapter includes the following sections:

- [Information About IPv4, page 2-1](#)
- [Licensing Requirements for IPv4, page 2-7](#)
- [Prerequisites for IPv4, page 2-7](#)
- [Guidelines and Limitations, page 2-7](#)
- [Default Settings, page 2-7](#)
- [Configuring IPv4, page 2-7](#)
- [Verifying the IPv4 Configuration, page 2-19](#)
- [Configuration Examples for IPv4, page 2-19](#)
- [Additional References, page 2-19](#)

Information About IPv4

You can configure IP on the switch 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 switch. An interface can have one primary IP address and multiple secondary addresses. All networking switches on an interface should share the same primary IP address because the packets that are generated by the switch always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address. 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 in the Cisco NX-OS system is responsible for handling IPv4 packets, as well as the forwarding of IPv4 packets, which includes IPv4 unicast route lookup, reverse path forwarding (RPF) checks, 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 and receive interface for IP clients.

This section includes the following topics:

- [Multiple IPv4 Addresses, page 2-2](#)
- [Address Resolution Protocol, page 2-3](#)
- [ARP Caching, page 2-3](#)
- [Static and Dynamic Entries in the ARP Cache, page 2-3](#)
- [Devices That Do Not Use ARP, page 2-4](#)
- [Reverse ARP, page 2-4](#)
- [Proxy ARP, page 2-5](#)
- [Local Proxy ARP, page 2-5](#)
- [Gratuitous ARP, page 2-5](#)
- [Glean Throttling, page 2-5](#)
- [ICMP, page 2-6](#)
- [ICMP Unreachable Support to Set Source Interface, page 2-6](#)
- [Virtualization Support, page 2-6](#)
- [IPv4 Routes with ECMP, page 2-6](#)

Multiple IPv4 Addresses

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

- When there are not enough host IP addresses for a particular network interface. For example, if your subnet 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 using 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 switch on a network segment uses a secondary IPv4 address, all other switches 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.

Address Resolution Protocol

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

Each switch compares the IP address to its own. Only the switch with the matching IP address replies to the switch that sends the data with a packet that contains the MAC address for the switch. The source switch adds the destination switch 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 switch lies on a remote network which is beyond another switch, the process is the same except that the switch 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 switch on the destination switch network uses ARP to obtain the MAC address of the destination switch and delivers the packet. ARP is enabled by default.

The default system-defined CoPP policy rate-limits ARP broadcast packets. 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 (switch) 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 memory for a period of time, which minimizes the use of valuable network resources to broadcast for the same address each time a packet is sent. You must maintain the cache entries since the cache entries are set to expire periodically because the information might become outdated. Every switch on a network updates its tables as addresses are broadcast.

Static and Dynamic Entries in the ARP Cache

You must manually configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each switch when using static routes. Static routing enables more control but 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 switches 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 that uses MAC addresses only, as opposed to a switch, which has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

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

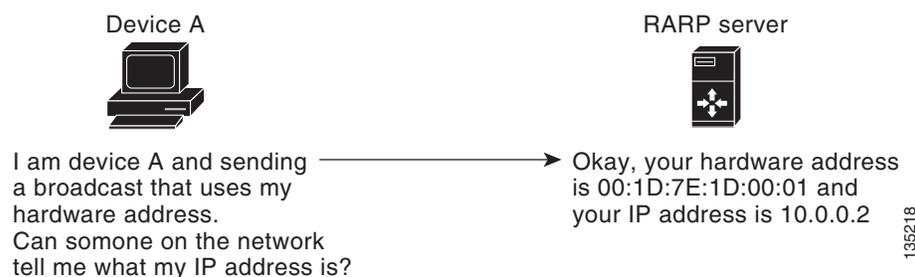
Layer 2 switches determine which port is connected to a device to which the message is addressed and send only to that port, unlike a hub, which sends the message out all of its ports. However, Layer 3 switches are switches that build an ARP cache (table).

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](#) illustrates how RARP works.

Figure 2-2 Reverse ARP



There are several limitations of RARP. 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 switch that is physically located on one network appear to be logically part of a different physical network connected to the same switch or firewall. Proxy ARP allows you to hide a switch with a public IP address on a private network behind a router and still have the switch 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 switches on a subnet reach remote subnets without configuring routing or a default gateway.

When switches 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 switches 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 switch and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The switch responds as if it is the remote destination for which the broadcast is addressed, with an ARP response that associates the MAC address of the switch with the IP address of the remote destination. The local switch 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 switch. By default, Proxy ARP is disabled.

Local Proxy ARP

You can use local Proxy ARP to enable a switch 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 switch to which they are connected.

Gratuitous ARP

Gratuitous ARP sends a request with identical source IP address and destination IP address to detect duplicate IP addresses. Cisco NX-OS supports enabling or disabling gratuitous ARP requests or ARP cache updates.

Glean Throttling

When forwarding an incoming IP packet, if the Address Resolution Protocol (ARP) request for the next-hop is not resolved, packets are punted to the central processing unit (CPU) for ARP resolution. The CPU resolves the MAC address for the next-hop and programs the hardware.

The Cisco Nexus 3000 Series device hardware has glean rate limiters to protect the supervisor from the glean traffic. If the maximum number of entries is exceeded, the packets for which the ARP request is not resolved continues to be processed in the software instead of getting dropped in 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.

ICMP

You can use 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.

ICMP Unreachable Support to Set Source Interface

You can configure an interface IP address for the ICMP source IP field to handle ICMP error messages. When ICMP packets are constructed in a network stack, the packets use the configured interface IP address. You can select Ethernet, loopback, or port channel interfaces to configure the IP address. If you do not select these options, the switch will behave as in earlier releases.

Virtualization Support

IPv4 supports Virtual Routing and Forwarding instances (VRFs). By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

IPv4 Routes with ECMP

If all next-hops for a route are glean, drop, or punt, all next-hops are programmed as-is in the Multipath hardware table.

If some next-hops for a route are glean, drop, or punt, and the remaining next-hops are not, then only non glean, drop, or punt next-hops are programmed in the Multipath hardware table.

When a specific next-hop for ECMP route is resolved (ARP ND resolved), then the Multipath hardware table is updated accordingly.

Licensing Requirements for IPv4

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	IPv4 requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images 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

IPv4 has the following configuration guidelines and limitations:

- You can configure a secondary IP address only after you configure the primary IP address.
- If the Cisco Nexus 3000 switch is used as a Layer 2 or Layer 3 termination switch, Cisco recommends that you set the **mac-address-table-aging-time** to 1800 (higher than the default ARP aging time of 1500 seconds) on all VLANs.
- The Cisco Nexus 3000 switch does not support per-VLAN cam aging timers.

Default Settings

Table 2-1 lists the default settings for IP parameters.

Table 2-1 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 a Static ARP Entry, page 2-10](#)
- [Configuring Proxy ARP, page 2-11](#)

- [Configuring Local Proxy ARP, page 2-12](#)
- [Configuring Gratuitous ARP, page 2-13](#)
- [Configuring IP Directed Broadcasts, page 2-14](#)
- [Configuring IP Glean Throttling, page 2-15](#)
- [Configuring the Hardware IP Glean Throttle Maximum, page 2-16](#)
- [Configuring a Hardware IP Glean Throttle Timeout, page 2-17](#)

**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. **no switchport**
4. **ip address** *ip-address/length* [**secondary**]
5. (Optional) **show ip interface**
6. (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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.

	Command	Purpose
Step 4	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.2.1.1 255.0.0.0	Specifies a primary or secondary IPv4 address for an interface. <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 - a 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 5	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to assign an IPv4 address:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip address 192.2.1.1 255.0.0.0
switch(config-if)# copy running-config startup-config
```

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. **no switchport**
4. **ip address** *ip-address/length* [secondary]
5. (Optional) **show ip interface**
6. (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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip address <i>ip-address/length</i> [<i>secondary</i>] Example: switch(config-if)# ip address 192.2.1.1 255.0.0.0 secondary	Specifies the configured address as a secondary IPv4 address.
Step 5	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring a Static ARP Entry

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

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **no switchport**
4. **ip arp *ipaddr mac_addr***
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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip arp <i>ipaddr mac_addr</i> Example: switch(config-if)# ip arp 192.2.1.1 0019.076c.1a78	Associates an IP address with a MAC address as a static entry.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a static ARP entry:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip arp 192.2.1.1 0019.076c.1a78
switch(config-if)# copy running-config startup-config
```

Configuring Proxy ARP

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

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip proxy-arp**
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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip proxy-arp Example: switch(config-if)# ip proxy-arp	Enables Proxy ARP on the interface.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure Proxy ARP:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip proxy-arp
switch(config-if)# copy running-config startup-config
```

Configuring Local Proxy ARP

You can configure Local Proxy ARP on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip local-proxy-arp**
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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip local-proxy-arp Example: switch(config-if)# ip local-proxy-arp	Enables Local Proxy ARP on the interface.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure Local Proxy ARP:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip local-proxy-arp
switch(config-if)# copy running-config startup-config
```

Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip arp gratuitous {request | update}**
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	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip arp gratuitous {request update} Example: switch(config-if)# ip arp gratuitous request	Enables gratuitous ARP on the interface. Default is enabled.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to disable gratuitous ARP requests:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# no ip arp gratuitous request
switch(config-if)# copy running-config startup-config
```

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 switch that is not directly connected to its destination subnet forwards an IP directed broadcast in the same way it would forward unicast IP packets destined to a host on that subnet. When a directed broadcast packet reaches a switch that is directly connected to its destination subnet, that packet is “exploded” as a 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 will be exploded as broadcasts on that subnet.

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

Command	Purpose
<code>ip directed-broadcast</code>	Enables the translation of a directed broadcast to physical broadcasts

Configuring IP Glean Throttling

Cisco NX-OS software supports glean throttling rate limiters to protect the supervisor from the glean traffic.



Note

We recommend that you configure the IP glean throttle feature by using the **hardware ip glean throttle** command 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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>hardware ip glean throttle</code> Example: switch(config)# <code>hardware ip glean throttle</code>	Enables ARP throttling.

	Command	Purpose
Step 3	no hardware ip glean throttle Example: switch(config)# no hardware ip glean throttle	Disables ARP throttling.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable IP glean throttling:

```
switch# configure terminal
switch(config)# hardware ip glean throttle
switch(config-if)# copy running-config startup-config
```

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).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **hardware ip glean throttle maximum *count***
3. **no hardware ip glean throttle maximum *count***
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	hardware ip glean throttle maximum <i>count</i> Example: switch(config)# hardware ip glean throttle maximum 2134	Configures the number of drop adjacencies that are installed in the FIB.

	Command	Purpose
Step 3	<pre>no hardware ip glean throttle maximum count</pre> <p>Example: switch(config)# no hardware ip glean throttle maximum 2134 </p>	<p>Applies the default limits. The default value is 1000. The range is from 0 to 16,383 entries.</p>
Step 4	<pre>copy running-config startup-config</pre> <p>Example: switch(config)# copy running-config startup-config </p>	(Optional) Saves this configuration change.

This example shows how to limit the maximum number of drop adjacencies that are installed in the FIB:

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum 2134
switch(config-if)# copy running-config startup-config
```

Configuring a Hardware IP Glean Throttle Timeout

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

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the `switchto vdc` command).

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	<pre>configure terminal</pre> <p>Example: switch# configure terminal switch(config)# </p>	Enters global configuration mode.
Step 2	<pre>hardware ip glean throttle maximum timeout timeout-in-sec</pre> <p>Example: switch(config)# hardware ip glean throttle maximum timeout 300 </p>	Configures the timeout for the installed drop adjacencies to remain in the FIB.

	Command	Purpose
Step 3	<pre>no hardware ip glean throttle maximum timeout timeout-in-sec</pre> <p>Example: <pre>switch(config)# no hardware ip glean throttle maximum timeout 300</pre></p>	<p>Applies the default limits.</p> <p>The timeout value is in seconds. The range is from 300 seconds (5 minutes) to 1800 seconds (30 minutes).</p> <p>Note After the timeout period is exceeded, the drop adjacencies are removed from the FIB.</p>
Step 4	<pre>copy running-config startup-config</pre> <p>Example: <pre>switch(config)# copy running-config startup-config</pre></p>	(Optional) Saves this configuration change.

This example shows how to configure a timeout for the drop adjacencies that are installed.

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum timeout 300
switch(config-if)# copy running-config startup-config
```

Configuring the Interface IP Address for the ICMP Source IP Field

You can configure an interface IP address for 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. **exit**

DETAILED STEPS

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

This example shows how to configure an interface IP address for the ICMP source IP field:

```
switch# configure terminal
switch(config)# ip source ethernet 1/1 icmp-errors
switch(config-if)# exit
```

This example shows how to unprovision an interface IP address from the ICMP source IP field:

```
switch# configure terminal
switch(config)# no ip source ethernet 1/1 icmp-errors
switch(config-if)# exit
```

Verifying the IPv4 Configuration

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

Command	Purpose
<code>show hardware forwarding ip verify</code>	Displays the IP packet verification configuration.
<code>show ip adjacency</code>	Displays the adjacency table.
<code>show ip arp</code>	Displays the ARP table.
<code>show ip interface</code>	Displays IP-related interface information.
<code>show ip arp statistics [vrf vrf-name]</code>	Displays the ARP statistics.
<code>show ip adjacency summary</code>	Displays the summary of number of throttle adjacencies.
<code>show ip arp summary</code>	Displays the summary of the number of throttle adjacencies.
<code>show ip adjacency throttle statistics</code>	Displays only the throttled adjacencies.

Configuration Examples for IPv4

This example shows how to configure an IPv4 address:

```
configure terminal
interface ethernet 1/2
no switchport
ip address 192.2.1.1/16
```

Additional References

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

- [Related Documents, page 2-20](#)
- [Standards, page 2-20](#)

Related Documents

Related Topic	Document Title
IP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

Standards

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