CHAPTER 2

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:

Information 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, reverse path forwarding (RPF) checks, and software access control list/policy-based routing (ACL/PBR) 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

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.

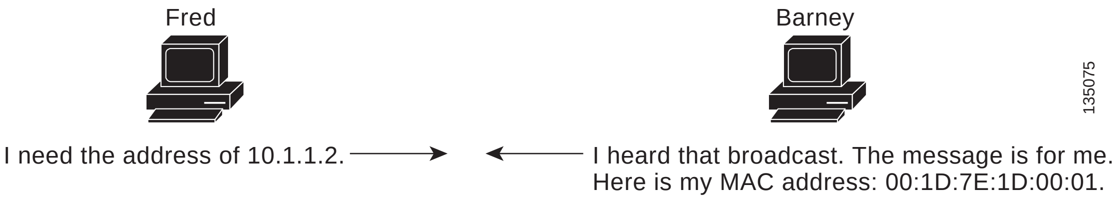
Note If any device on a network segment uses a secondary IPv4 address, all other devices on that same network interface must also use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause routing loops.

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


Fred
Barney
135075
I need the address of 10.1.1.2.
I heard that broadcast. The message is for me.
Here is my MAC address: 00:1D:7E:1D:00:01.

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.

Note Cisco Nexus 7000 Series devices do not support Ethernet SNAP encoding.

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 is connected to a device to which the message is addressed and 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 illustrates 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:

- Since 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 Release 4.0(3) and later releases support enabling or disabling gratuitous ARP requests or ARP cache updates.

Glean Throttling

When forwarding an incoming IP packet in a line card, if the Address Resolution Protocol (ARP) request for the next hop is not resolved, 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.

The Cisco Nexus 7000 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.

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.

Note Please ensure you enable **ip unreachables** command between TCP endpoints for the Path MTU discovery feature to work correctly.

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. VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Cisco NX-OS Virtual Device Context Configuration Guide* and see Chapter 14, “Configuring Layer 3 Virtualization.”

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 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 *Cisco NX-OS Licensing Guide*.

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.
- Cisco Nexus 7000 Series devices do not support Ethernet SNAP encoding.

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 Path MTU Discovery, page 2-14
- Configuring IP Packet Verification, page 2-15
- Configuring IP Directed Broadcasts, page 2-16

- Configuring IP Glean Throttling, page 2-17
- Configuring the Hardware IP Glean Throttle Maximum, page 2-18
- Configuring a Hardware IP Glean Throttle Timeout, page 2-18
- Configuring the Hardware IP Glean Throttle Syslog, page 2-19

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.

BEFORE YOU BEGIN

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

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 configuration mode.

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

	Command	Purpose

Step 3	**`ip address`** *`ip-address/length`* `[secondary]`
	Example:
	`switch(config-if)# ip address 192.168.1.1 255.0.0.0`
		Specifies a primary or secondary IPv4 address for an interface.
		• 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 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) 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)# ip address 192.168.1.1 255.0.0.0
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.

BEFORE YOU BEGIN

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

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 configuration mode.

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

Step 3	**`ip address`** *`ip-address/length`* `[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) Saves this configuration change.

Configuring 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.

BEFORE YOU BEGIN

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

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 configuration mode.

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

Step 3	**`ip arp`** *`ipaddr mac_addr`*
	
	`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) 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)# ip arp 192.168.1.1 0019.076c.1a78
switch(config-if)# copy running-config startup-config
```

Configuring Proxy ARP

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

BEFORE YOU BEGIN

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

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 configuration mode.

Step 2	**interface ethernet** *number*
	
	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) Saves this configuration change.

This example shows how to configure Proxy ARP:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
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 device.

BEFORE YOU BEGIN

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

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 configuration mode.

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

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

Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

BEFORE YOU BEGIN

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

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 configuration mode.
Step 2	**interface ethernet** *number* **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) 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 ip arp gratuitous request
switch(config-if)# copy running-config startup-config
```

Configuring Path MTU Discovery

You can configure path MTU discovery.

BEFORE YOU BEGIN

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

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 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) Saves this configuration change.

Configuring IP Packet Verification

Cisco NX-OS supports an Intrusion Detection System (IDS) that checks for IP packet verification. You can enable or disable these IDS checks.

To enable IDS checks, use the following commands in global configuration mode:

Command	Purpose
hardware ip verify address {**destination zero** | **identical** | **reserved** | **source** {**broadcast** | **multicast**}}	Performs the following IDS checks on the IP address: • **destination zero**—Drops IP packets if the destination IP address is 0.0.0.0. • **identical**—Drops IP packets if the source IP address is identical to the destination IP address. • **reserved**—Drops IP packets if the IP address is in the 127.x.x.x range. • **source**—Drops IP packets if the IP source address is either 255.255.255.255 (broadcast) or in the 224.x.x.x range (multicast).
hardware ip verify checksum	Drops IP packets if the packet checksum is invalid.
hardware ip verify fragment	Drops IP packets if the packet fragment has a nonzero offset and the DF bit is active.

Command	Purpose
hardware ip verify length {**consistent** | **maximum** {**max-frag** | **max-tcp** | **udp**} | **minimum**}	Performs the following IDS checks on the IP address: • **consistent**—Drops IP packets where the Ethernet frame size is greater than or equal to the IP packet length plus the Ethernet header. • **maximum max-frag**—Drops IP packets if the maximum fragment offset is greater than 65536. • **maximum max-tcp**—Drops IP packets if the TCP length is greater than the IP payload length. • **maximum udp**—Drops IP packets if the IP payload length is less than the UDP packet length. • **minimum**—Drops IP packets if the Ethernet frame length is less than the IP packet length plus four octets (the CRC length).
hardware ip verify tcp tiny-frag	Drops TCP packets if the IP fragment offset is 1, or if the IP fragment offset is 0 and the IP payload length is less than 16.
hardware ip verify version	Drops IP packets if the ethertype is not set to 4 (IPv4).

Use the **show hardware forwarding ip verify** command to display the IP packet verification 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 broacasts 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 [*acl*]	Enables the translation of a directed broadcast to physical broadcasts. You can optionally filter those broacasts through an IP access list.

Configuring IP Glean Throttling

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

You can enable IP glean throttling.

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

Step 2	**hardware ip glean throttle** **Example:** `switch(config)# hardware ip glean throttle`	Enables ARP throttling.
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 configuration mode.

Step 2	**hardware ip glean throttle maximum** ***count*** **Example:** `switch(config)# hardware ip glean throttle maximum 2134`	Configures the number of drop adjacencies that are installed in the FIB.
Step 3	no **hardware ip glean throttle maximum** ***count*** **Example:** `switch(config)# no hardware ip glean throttle maximum 2134`	Applies the default limits. The default value is 1000. The range is from 0 to 32767 entries.
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 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	**`configure terminal`**
	`Example:`
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**`hardware ip glean throttle maximum timeout`** ***`timeout-in-sec`***
	`Example:`
	`switch(config)# hardware ip glean throttle maximum timeout 300`
		Configures the timeout for the installed drop adjacencies to remain in the FIB.

Step 3	`no` **`hardware ip glean throttle maximum timeout`** ***`timeout-in-sec`***
	`Example:`
	`switch(config)# no hardware ip glean throttle maximum timeout 300`
		Applies the default limits.
		The timeout value is in seconds. 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 4	**`copy running-config startup-config`**
	`Example:`
	`switch(config)# copy running-config startup-config`
		(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 Hardware IP Glean Throttle Syslog

You can generate a syslog if the number of packets that get dropped for a specific flow exceeds the configured packet count.

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 syslog** *pck-count*
3. **no hardware ip glean throttle syslog** *pck-count*
4. **(Optional) copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters configuration mode.
Step 2	**hardware ip glean throttle syslog** *pck-count* **Example:** switch(config)# hardware ip glean throttle syslog 1030	Generates a syslog if the number of packets that get dropped for a specific flow exceed the configured packet count.
Step 3	no **hardware ip glean throttle syslog** *pck-count* **Example:** switch(config)# no hardware ip glean throttle syslog 1030	Applies the default limits. The default is 10000 packets. The range is from 0 to 65535 packets. **Note** After the timeout period is exceeded, the drop adjacencies are removed from the FIB.
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 generate a syslog if the number of packets that get dropped for a specific flow exceeds the configured packet count:

```
switch# configure terminal
switch(config)# hardware ip glean throttle syslog 1030
switch(config-if)# copy running-config startup-config
```

Verifying the IPv4 Configuration

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

Command	Purpose
show hardware forwarding ip verify	Displays the IP packet verification configuration.
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays the summary of number of throttle adjacencies.

Command	Purpose
show ip arp	Displays the ARP table.
show ip arp summary	Displays the summary of the number of throttle adjacencies.
show ip adjacency throttle statistics	Displays only the throttled adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [**vrf** *vrf-name*]	Displays the ARP statistics.

Configuration Examples for IPv4

The N7K-F132-15 module only runs Layer 2 switching. So, when you have both this module and an M Series module in one Nexus 7000 Series chassis and you are performing Layer 3 procedures, the system uses proxy routing. You can also configure proxy routing.

This section includes the following topics:

- Example: Reserving All Ports on a Module for Proxy Routing, page 2-21
- Example: Reserving Ports for Proxy Routing, page 2-23
- Example: Excluding Ports From Proxy Routing, page 2-24

Example: Reserving All Ports on a Module for Proxy Routing

This example shows how to reserve all ports on a module for proxy routing.

Step 1 Determine which modules are present in the device.

```
switch# show module
Mod  Ports  Module-Type                      Model              Status
---  -----  -------------------------------- ------------------ ------------
1    32     10 Gbps Ethernet Module          N7K-M132XP-12      ok
2    48     10/100/1000 Mbps Ethernet Module N7K-M148GT-11      ok
3    48     1000 Mbps Optical Ethernet Modul N7K-M148GS-11      ok
5    0      Supervisor module-1X             N7K-SUP1           active *
6    0      Supervisor module-1X             N7K-SUP1           ha-standby
8    32     1/10 Gbps Ethernet Module        N7K-F132XP-15      ok
```

The F1 module is in Slot 8, and the M1 modules are in Slots 1 - 3.

Step 2 Determine which ports are available in the VDC.

```
switch# show vdc membership | end "Ethernet3/48"

vdc_id: 0 vdc_name: Unallocated interfaces:

vdc_id: 1 vdc_name: switch interfaces:
       Ethernet1/9           Ethernet1/10          Ethernet1/11
       Ethernet1/12          Ethernet1/13          Ethernet1/14
       Ethernet1/15          Ethernet1/16          Ethernet1/17
       Ethernet1/18          Ethernet1/19          Ethernet1/20
       Ethernet1/21          Ethernet1/22          Ethernet1/23
       Ethernet1/24          Ethernet1/25          Ethernet1/26
       Ethernet1/27          Ethernet1/28          Ethernet1/29
       Ethernet1/30          Ethernet1/31          Ethernet1/32
       Ethernet2/1           Ethernet2/2           Ethernet2/3
       Ethernet2/4           Ethernet2/5           Ethernet2/6
       Ethernet2/7           Ethernet2/8           Ethernet2/9
       Ethernet2/10          Ethernet2/11          Ethernet2/12
       Ethernet2/25          Ethernet2/26          Ethernet2/27
       Ethernet2/28          Ethernet2/29          Ethernet2/30
       Ethernet2/31          Ethernet2/32          Ethernet2/33
       Ethernet2/34          Ethernet2/35          Ethernet2/36
       Ethernet2/37          Ethernet2/38          Ethernet2/39
       Ethernet2/40          Ethernet2/41          Ethernet2/42
       Ethernet2/43          Ethernet2/44          Ethernet2/45
       Ethernet2/46          Ethernet2/47          Ethernet2/48

       Ethernet3/1           Ethernet3/2           Ethernet3/3
       Ethernet3/4           Ethernet3/5           Ethernet3/6
       Ethernet3/7           Ethernet3/8           Ethernet3/9
       Ethernet3/10          Ethernet3/11          Ethernet3/12
       Ethernet3/13          Ethernet3/14          Ethernet3/15
       Ethernet3/16          Ethernet3/17          Ethernet3/18
       Ethernet3/19          Ethernet3/20          Ethernet3/21
       Ethernet3/22          Ethernet3/23          Ethernet3/24
       Ethernet3/25          Ethernet3/26          Ethernet3/27
       Ethernet3/28          Ethernet3/29          Ethernet3/30
       Ethernet3/31          Ethernet3/32          Ethernet3/33
       Ethernet3/34          Ethernet3/35          Ethernet3/36
       Ethernet3/37          Ethernet3/38          Ethernet3/39
       Ethernet3/40          Ethernet3/41          Ethernet3/42
       Ethernet3/43          Ethernet3/44          Ethernet3/45
       Ethernet3/46          Ethernet3/47          Ethernet3/48
```

Step 3 Determine which ports are available for proxy routing.

```
switch# show hardware proxy layer-3 detail

Global Information:
       F1 Modules:     Count: 1         Slot: 8
       M1 Modules:     Count: 3         Slot: 1-3

       Replication Rebalance Mode:             Manual
       Number of proxy layer-3 forwarders:     13
       Number of proxy layer-3 replicators:    8

Forwarder Interfaces                     Status     Reason
------------------------------------------------------------------------------
Eth1/9, Eth1/11, Eth1/13, Eth1/15        up         SUCCESS
Eth1/10, Eth1/12, Eth1/14, Eth1/16       up         SUCCESS
Eth1/17, Eth1/19, Eth1/21, Eth1/23       up         SUCCESS
Eth1/18, Eth1/20, Eth1/22, Eth1/24       up         SUCCESS
Eth1/25, Eth1/27, Eth1/29, Eth1/31       up         SUCCESS
Eth1/26, Eth1/28, Eth1/30, Eth1/32       up         SUCCESS
Eth2/1-12                                up         SUCCESS
Eth2/25-36                               up         SUCCESS
Eth2/37-48                               up         SUCCESS
Eth3/1-12                                up         SUCCESS
Eth3/13-24                               up         SUCCESS
Eth3/25-36                               up         SUCCESS
Eth3/37-48                               up         SUCCESS

Replicator Interfaces                    #Interface-Vlan   Interface-Vlan
------------------------------------------------------------------------------
Eth1/1, Eth1/3, Eth1/5, Eth1/7, Eth1/9,  0
Eth1/11, Eth1/13, Eth1/15
Eth1/2, Eth1/4, Eth1/6, Eth1/8, Eth1/10, 0
```

```
Eth1/12, Eth1/14, Eth1/16
Eth1/17, Eth1/19, Eth1/21, Eth1/23,      0
Eth1/25, Eth1/27, Eth1/29, Eth1/31
Eth1/18, Eth1/20, Eth1/22, Eth1/24,      0
Eth1/26, Eth1/28, Eth1/30, Eth1/32
Eth2/1-24                                0
Eth2/25-48                               0
Eth3/1-24                                0
Eth3/25-48                               0
switch#
```

Note Ports are listed in their respective port-groups.

Step 4 Reserve a module for unicast and multicast proxy routing.

```
switch# configure terminal
switch(config)# hardware proxy layer-3 forwarding use module 2
switch(config)# hardware proxy layer-3 replication use module 2
```

Step 5 Verify this configuration.

```
switch(config)# show hardware proxy layer-3 detail

Global Information:
       F1 Modules:     Count: 1         Slot: 8
       M1 Modules:     Count: 3         Slot: 1-3

       Replication Rebalance Mode:             Manual
       Number of proxy layer-3 forwarders:     3
       Number of proxy layer-3 replicators:    2

Forwarder Interfaces                     Status     Reason
------------------------------------------------------------------------------
Eth2/1-12                                up         SUCCESS
Eth2/25-36                               up         SUCCESS
Eth2/37-48                               up         SUCCESS

Replicator Interfaces                    #Interface-Vlan   Interface-Vlan
------------------------------------------------------------------------------
Eth2/1-24                                0
Eth2/25-48                               0
switch(config)#
```

Example: Reserving Ports for Proxy Routing

This example shows how to reserve some ports on a module for proxy routing.

Step 1 Reserve a subset of ports on a module.

```
switch(config)# hardware proxy layer-3 forwarding use interface ethernet 2/1-6 <----
-subset of port group
switch(config)# hardware proxy layer-3 replication use interface ethernet 2/1-6 <----
-subset of port group
```

This example reserves a subset of ports from a port group.

Step 2 Verify this configuration.

```
switch(config)# show hardware proxy layer-3 detail
```

```
Global Information:
       F1 Modules:     Count: 1         Slot: 8
       M1 Modules:     Count: 3         Slot: 1-3

       Replication Rebalance Mode:             Manual
       Number of proxy layer-3 forwarders:     1
       Number of proxy layer-3 replicators:    1

Forwarder Interfaces                     Status     Reason
-------------------------------------------------------------------------------
Eth2/1-12                                up         SUCCESS

Replicator Interfaces                    #Interface-Vlan   Interface-Vlan
-------------------------------------------------------------------------------
Eth2/1-24                                0 <----------- full port group
switch(config)#
```

Note All ports in a port group are reserved for proxy routing.

Example: Excluding Ports From Proxy Routing

This example shows how to exclude some ports on a module from proxy routing.

Step 1 Exclude a subset of ports on a module.

```
switch(config)# hardware proxy layer-3 forwarding exclude interface ethernet 2/1-12
<---subset of port group
switch(config)# hardware proxy layer-3 replication exclude interface ethernet 2/1-12
```

Step 2 Verify this configuration.

```
switch(config)# show hardware proxy layer-3 detail

Global Information:
       F1 Modules:     Count: 1         Slot: 8
       M1 Modules:     Count: 3         Slot: 1-3

       Replication Rebalance Mode:             Manual
       Number of proxy layer-3 forwarders:     12
       Number of proxy layer-3 replicators:    7

Forwarder Interfaces                     Status     Reason
-------------------------------------------------------------------------------
Eth1/9, Eth1/11, Eth1/13, Eth1/15        up         SUCCESS
Eth1/10, Eth1/12, Eth1/14, Eth1/16       up         SUCCESS
Eth1/17, Eth1/19, Eth1/21, Eth1/23       up         SUCCESS
Eth1/18, Eth1/20, Eth1/22, Eth1/24       up         SUCCESS
Eth1/25, Eth1/27, Eth1/29, Eth1/31       up         SUCCESS
Eth1/26, Eth1/28, Eth1/30, Eth1/32       up         SUCCESS
Eth2/25-36                               up         SUCCESS
Eth2/37-48                               up         SUCCESS
Eth3/1-12                                up         SUCCESS
Eth3/13-24                               up         SUCCESS
Eth3/25-36                               up         SUCCESS
Eth3/37-48                               up         SUCCESS
```

```
Replicator Interfaces                    #Interface-Vlan   Interface-Vlan
------------------------------------------------------------------------------
Eth1/1, Eth1/3, Eth1/5, Eth1/7, Eth1/9,  0
Eth1/11, Eth1/13, Eth1/15
Eth1/2, Eth1/4, Eth1/6, Eth1/8, Eth1/10, 0
Eth1/12, Eth1/14, Eth1/16
Eth1/17, Eth1/19, Eth1/21, Eth1/23,      0
Eth1/25, Eth1/27, Eth1/29, Eth1/31
Eth1/18, Eth1/20, Eth1/22, Eth1/24,      0
Eth1/26, Eth1/28, Eth1/30, Eth1/32
Eth2/25-48                               0 <---- e 2/1-24 excluded
Eth3/1-24                                0
Eth3/25-48                               0
switch(config)#
```

Note All ports in the port group are excluded from proxy routing.

Additional References

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

- Related Documents, page 2-26
- Standards, page 2-26

Related Documents

Related Topic	Document Title
IP CLI commands	*Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference, Release 5.x*

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.	—

Feature History for IP

Table 2-2 lists the release history for this feature.

Table 2-2 Feature History for IP

Feature Name	Releases	Feature Information
ACL filter for IP directed broadcasts	5.2(1)	Added support to filter IP directed broadcasts through an IP access list.
Glean Throttling	5.1(1)	Added support for IPv4 glean throttling.
ARP	4.1(4)	Added support to protect against an ARP broadcast storm.
IP	4.1(3)	Changed the **platform ip verify** command to the **hardware ip verify** command.
ARP	4.0(3)	Added support for gratuitous ARP. The following command was added: • **ip arp gratuitous** {**request** | **update**}
IP	4.0(1)	This feature was introduced.