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:

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.

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 and multicast 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.

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. The following figure shows the ARP broadcast and response process.

Figure 1: ARP Process

When the destination switch lies on a remote network which is beyond another router, 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 data 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.

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.

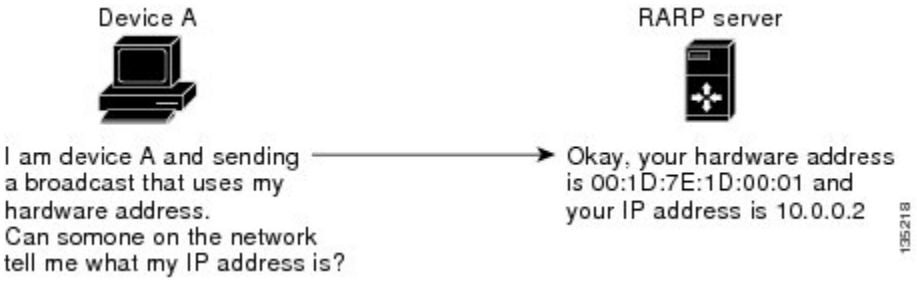
Layer 2 switches determine which port is connected to a device to which the message is addressed and send only to that port, unlike a hub, which sends the message out all of its ports. However, Layer 3 switches are switches that build an ARP cache (table).

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. The following figure illustrates how RARP works.

Figure 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 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.

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

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.

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 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.

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 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 switch does not support per-VLAN cam aging timers.

Default Settings

The following table lists the default settings for IP parameters.

Table 1: Default IP Parameters

Parameters	Default
ARP timeout	1500 seconds
Proxy ARP	Disabled

Configuring IPv4

Configuring IPv4 Addressing

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

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet number**
	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** *ip-address* / *length* [**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. • 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	(Optional) **show ip interface** **Example:** `switch(config-if)# show ip interface`	Displays interfaces configured for IPv4.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Example

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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet number**
	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** *ip-address / length* [**secondary**] **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	(Optional) **show ip interface** **Example:** `switch(config-if)# show ip interface`	Displays interfaces configured for IPv4.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet number**
	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 ipaddr mac_addr** **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`	Saves this configuration change.

Example

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 1 92.2.1.1 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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet number**
	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	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Example

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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet number**
	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	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Example

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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

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

Example

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 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
ip directed-broadcast	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.

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

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global 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	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Example

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 Timeout

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

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hardware ip glean throttle maximum timeout** *timeout-in-seconds* **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-seconds* **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	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to disable gratuitous ARP requests:

```
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 the ICMP source IP field to handle ICMP error messages.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip source** {**ethernet** *slot/port* | **loopback** *number* | **port-channel** *number*} {**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.

Example

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
```

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

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

Configuring Logging for Software Forwarding of IP Packets

You can configure the logging conditions for IP packets that are forwarded by the NX-OS software. The conditions consist of the following:

- A minimum number of packets (the size)
- An optional period of time (the logging interval)

The logging conditions create the packet per second (pps) threshold. When traffic meets or exceeds the conditions, NX-OS logs a console message. For example:

```
2019 jul 31 15:28:31 switch-1 %$ VDC-1 %$ %USER-3-SYSTEM_MSG: Packets per second exceeded
the configured threshold 40, current PPS: 1262  - netstack
```

You can set the conditions for forwarded packets through the **ip pps threshold unicast-forward** command. To disable the feature, use **no ip pps threshold unicast-forward**.

Procedure

	Command or Action	Purpose
Step 1	**config terminal** **Example:**	Enters the configuration terminal.

	Command or Action	Purpose

	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`	

Step 2	**ip pps threshold unicast-forward** *pps-threshold* [*syslog-interval*]
	Example:
	`switch-1(config)# ip pps threshold unicast-forward 50 5`
	`switch-1(config)#`
		Enable the feature and set the conditions:
		• The *pps-threshold* is from 1 through 30000 packets.
		• The *syslog-interval* is from 1 through 60 seconds. The default is 1 second.

Step 3	(Optional) **show ip pps threshold**
	Example:
	`switch-1(config) show ip traffic pps`
	`PPS type : unicast-forward, PPS limit : 50, Log Interval: 5`
	`switch-1(config)#`
		Display the current PPS threshold configuration.

Example

This example shows how to configure a console message if the number of packets that get forwarded for a specific flow exceeds the configured packet count and a logging interval of 4000 packets every 2 seconds:

```
switch-1# configure terminal
switch-1(config)# ip pps threshold unicast-forward 4000 2
switch-1(config)# copy running-config startup-config
```

Verifying the IPv4 Configuration

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

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

Command	Purpose
show ip interface	Displays IP-related interface information.

Configuration Examples for IPv4

This example shows how to configure an IPv4 address:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config)# no switchport
switch(config-if)#ip address 192.2.1.1/1
```