



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 device.

This chapter includes the following sections:

- [Information About IPv4, page 2-1](#)
- [Licensing Requirements for IPv4, page 2-6](#)
- [Prerequisites for IPv4, page 2-6](#)
- [Guidelines and Limitations, page 2-6](#)
- [Configuring IPv4, page 2-6](#)
- [Verifying the IPv4 Configuration, page 2-14](#)
- [IPv4 Example Configuration, page 2-14](#)
- [Default Settings, page 2-14](#)
- [Additional References, page 2-14](#)
- [Feature History for IP, page 2-15](#)

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. 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.

Send document comments to nexus7k-docfeedback@cisco.com.

The IP feature in the Cisco NX-OS system 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, page 2-2](#)
- [Address Resolution Protocol, page 2-2](#)
- [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](#)
- [Reverse ARP, page 2-4](#)
- [Proxy ARP, page 2-5](#)
- [Local Proxy ARP, page 2-5](#)
- [ICMP, page 2-5](#)
- [Virtualization Support, 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 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 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 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.

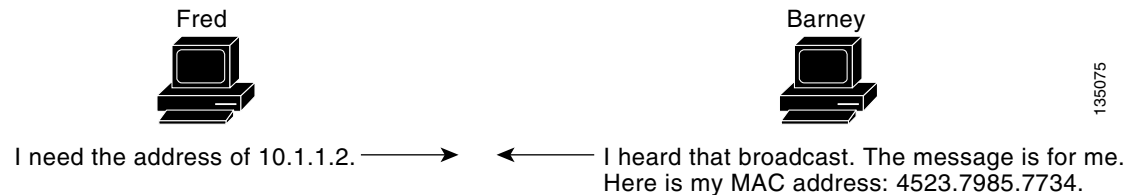
Address Resolution Protocol

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.

Send document comments to nexus7k-docfeedback@cisco.com.

Each device compares the IP address to its own. Only the device with the matching IP address replies to the device that sends the data with a packet that contains the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. Figure 2-1 shows the ARP broadcast and response process.

Figure 2-1 ARP Process



When the destination device lies on a remote network which 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.

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 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 device 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 device 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 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.

Send document comments to nexus7k-docfeedback@cisco.com.

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, as opposed to a device, which 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.

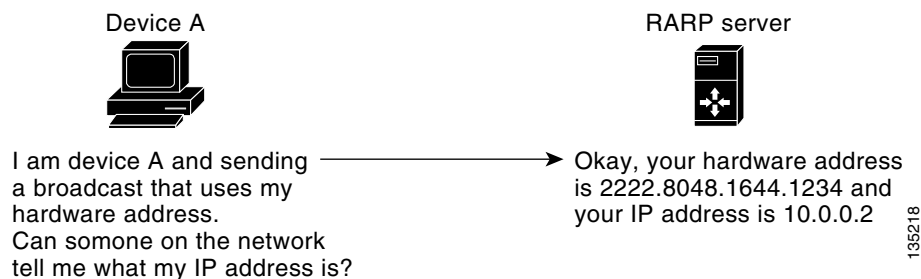
Layer 2 switches determine which port is connected to a device to which the message is addressed and send only to that port, unlike a hub, which sends the message out all its ports. However, Layer 3 switches are devices that build an ARP cache (table).

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



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:

- 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.

Send document comments to nexus7k-docfeedback@cisco.com.

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 identical source IP address and 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.

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

[Send document comments to nexus7k-docfeedback@cisco.com.](mailto:nexus7k-docfeedback@cisco.com)



Note

ICMP redirects are disabled on interfaces where the local proxy ARP feature is enabled.

Virtualization Support

IPv4 supports Virtual Routing and Forwarding instances (VRFs). 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 Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.0* 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
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 NX-OS licensing scheme, see the <i>Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.0</i> .

Prerequisites for IPv4

IPv4 has the following prerequisites:

- Can only be configured on Layer 3 interfaces.

Guidelines and Limitations

IPv4 has the following guidelines and limitations and restrictions:

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

Configuring IPv4

This section includes the following topics:

- [Configuring IPv4 Addressing, page 2-7](#)
- [Configuring Multiple IP Addresses, page 2-8](#)
- [Configuring a Static ARP Entry, page 2-9](#)
- [Configuring Proxy ARP, page 2-10](#)
- [Configuring Local Proxy ARP, page 2-11](#)
- [Configuring IP Packet Verification, page 2-12](#)

Send document comments to nexus7k-docfeedback@cisco.com.

**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. **config t**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length*
4. **show ip interface**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	config t Example: switch# config t switch(config)#	Enters configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.2.1.1 255.0.0.0	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.

Send document comments to nexus7k-docfeedback@cisco.com.

	Command	Purpose
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# config t
switch(config)# interface ethernet 2/3
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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **interface ethernet *number***
3. **ip address *ip-address/length***
4. **show ip interface**
5. **copy running-config startup-config**

DETAILED STEPS

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

Send document comments to nexus7k-docfeedback@cisco.com.

	Command	Purpose
Step 3	ip address <i>ip-address/length</i> [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 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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *number*
3. **ip arp** *ipaddr mac_addr*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	config t	Enters configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip arp <i>ipaddr mac_addr</i> Example: switch(config-if)# ip arp 192.2.1.1 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.

Send document comments to nexus7k-docfeedback@cisco.com.

This example shows how to configure a static ARP entry:

```
switch# config t
switch(config)# interface ethernet 2/3
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 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. **config t**
2. **interface ethernet *number***
3. **ip proxy-arp**
4. **copy running-config startup-config**

DETAILED STEPS

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

This example shows how to configure Proxy ARP:

```
switch# config t
switch(config)# interface ethernet 2/3
switch(config-if)# ip proxy-arp
switch(config-if)# copy running-config startup-config
```

Send document comments to nexus7k-docfeedback@cisco.com.

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. **config t**
2. **interface ethernet *number***
3. **ip local-proxy-arp**
4. **copy running-config startup-config**

DETAILED STEPS

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

Send document comments to nexus7k-docfeedback@cisco.com.

SUMMARY STEPS

1. **config t**
2. **interface ethernet *number***
3. **ip arp gratuitous {request | update}**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	config t	Enters configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip arp gratuitous {request update} Example: switch(config-if)# ip arp gratuitous request	Enables gratuitous ARP on the interface. 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# config t
switch(config)# interface ethernet 2/3
switch(config-if)# no ip arp gratuitous request
switch(config-if)# copy running-config startup-config
```

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.

Send document comments to nexus7k-docfeedback@cisco.com.

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

Command	Purpose
platform ip verify address { destination zero identical reserved source { broadcast multicast }}	Performs the following IDS checks on the IP address: <ul style="list-style-type: none"> • 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).
platform ip verify checksum	Drops IP packets if the packet checksum is invalid.
platform ip verify fragment	Drops IP packets if the packet fragment has a nonzero offset and the DF bit is active.
platform ip verify length { consistent maximum { max-frag max-tcp udp } minimum }	Performs the following IDS checks on the IP address: <ul style="list-style-type: none"> • 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).
platform 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.
platform 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.

[Send document comments to nexus7k-docfeedback@cisco.com.](mailto:nexus7k-docfeedback@cisco.com)

Verifying the IPv4 Configuration

To verify configuration information, use the following commands:

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.

IPv4 Example Configuration

This example shows how to configure an IPv4 address:

```
config t
interface e 1/2
no switchport
ip address 192.2.1.1/16
```

Default Settings

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

Table 2-1 Default IP Parameters

Parameters	Default
proxy ARP	disabled

Additional References

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

- [Related Documents, page 2-15](#)
- [Standards, page 2-15](#)

Send document comments to nexus7k-docfeedback@cisco.com.

Related Documents

Related Topic	Document Title
IP CLI commands	<i>Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference, Release 4.0</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.	—

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
ARP	4.0(3)	Added support for gratuitous ARP. The following command was added: <ul style="list-style-type: none"> ip arp gratuitous {request update}
IP	4.0(1)	This feature was introduced.

Send document comments to nexus7k-docfeedback@cisco.com.