



Configuring ARP

Address resolution is the process of mapping network addresses to Media Access Control (MAC) addresses. This process is accomplished using the Address Resolution Protocol (ARP).

**Note**

For a complete description of the ARP commands listed in this module, refer to the *IP Addresses and Services Command Reference for Cisco NCS 6000 Series Routers*. To locate documentation of other commands that appear in this module, use the command reference master index, or search online.

Feature History for Configuring ARP

Release	Modification
Release 5.0.0	This feature was introduced.

- [Prerequisites for Configuring ARP , page 1](#)
- [Restrictions for Configuring ARP , page 1](#)
- [Information About Configuring ARP , page 2](#)
- [How to Configure ARP , page 4](#)

Prerequisites for Configuring ARP

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for Configuring ARP

The following restrictions apply to configuring ARP :

- Reverse Address Resolution Protocol (RARP) is not supported.
- ARP throttling is not supported.

**Note**

ARP throttling is the rate limiting of ARP packets in Forwarding Information Base (FIB).

Information About Configuring ARP

To configure ARP, you must understand the following concepts:

IP Addressing Overview

A device in the IP can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is more properly known as a *data link address*, because it is contained in the data link layer (Layer 2 of the OSI model) part of the packet header and is read by data-link devices (bridges and all device interfaces, for example). The more technically inclined person will refer to local addresses as *MAC addresses*, because the MAC sublayer within the data link layer processes addresses for the layer.

To communicate with a device on Ethernet, for example, Cisco IOS XR software first must determine the 48-bit MAC or local data-link address of that device. The process of determining the local data-link address from an IP address is called *address resolution*.

Address Resolution on a Single LAN

The following process describes address resolution when the source and destination devices are attached to the same LAN:

- 1 End System A broadcasts an ARP request onto the LAN, attempting to learn the MAC address of End System B.
- 2 The broadcast is received and processed by all devices on the LAN, including End System B.
- 3 Only End System B replies to the ARP request. It sends an ARP reply containing its MAC address to End System A.
- 4 End System A receives the reply and saves the MAC address of End System B in its ARP cache. (The ARP cache is where network addresses are associated with MAC addresses.)
- 5 Whenever End System A needs to communicate with End System B, it checks the ARP cache, finds the MAC address of System B, and sends the frame directly, without needing to first use an ARP request.

Address Resolution When Interconnected by a Router

The following process describes address resolution when the source and destination devices are attached to different LANs that are interconnected by a router (only if proxy-arp is turned on):

- 1 End System Y broadcasts an ARP request onto the LAN, attempting to learn the MAC address of End System Z.
- 2 The broadcast is received and processed by all devices on the LAN, including Router X.
- 3 Router X checks its routing table and finds that End System Z is located on a different LAN.
- 4 Router X therefore acts as a proxy for End System Z. It replies to the ARP request from End System Y, sending an ARP reply containing its own MAC address as if it belonged to End System Z.
- 5 End System Y receives the ARP reply and saves the MAC address of Router X in its ARP cache, in the entry for End System Z.
- 6 When End System Y needs to communicate with End System Z, it checks the ARP cache, finds the MAC address of Router X, and sends the frame directly, without using ARP requests.
- 7 Router X receives the traffic from End System Y and forwards it to End System Z on the other LAN.

ARP and Proxy ARP

Two forms of address resolution are supported by Cisco IOS XR software: Address Resolution Protocol (ARP) and proxy ARP, as defined in RFC 826 and RFC 1027, respectively.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. After a media or MAC address is determined, the IP address or media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network.

When proxy ARP is disabled, the networking device responds to ARP requests received on an interface only if one of the following conditions is met:

- The target IP address in the ARP request is the same as the interface IP address on which the request is received.
- The target IP address in the ARP request has a statically configured ARP alias.

When proxy ARP is enabled, the networking device also responds to ARP requests that meet all the following conditions:

- The target IP address is not on the same physical network (LAN) on which the request is received.
- The networking device has one or more routes to the target IP address.
- All of the routes to the target IP address go through interfaces other than the one on which the request is received.

ARP Cache Entries

ARP establishes correspondences between network addresses (an IP address, for example) and Ethernet hardware addresses. A record of each correspondence is kept in a cache for a predetermined amount of time and then discarded.

You can also add a static (permanent) entry to the ARP cache that persists until expressly removed.

How to Configure ARP

This section contains instructions for the following tasks:

Defining a Static ARP Cache Entry

ARP and other address resolution protocols provide a dynamic mapping between IP addresses and media addresses. Because most hosts support dynamic address resolution, generally you need not to specify static ARP cache entries. If you must define them, you can do so globally. Performing this task installs a permanent entry in the ARP cache. Cisco IOS XR software uses this entry to translate 32-bit IP addresses into 48-bit hardware addresses.

Optionally, you can specify that the software responds to ARP requests as if it were the owner of the specified IP address by making an alias entry in the ARP cache.

SUMMARY STEPS

- 1. configure**
- 2. Do one of the following:**
 - **arp ip-address hardware-address encapsulation-type**
 - **arp ip-address hardware-address encapsulation-type alias**
- 3. commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	<p>Do one of the following:</p> <ul style="list-style-type: none"> • arp ip-address hardware-address encapsulation-type • arp ip-address hardware-address encapsulation-type alias <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa or RP/0/RP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa alias</pre>	<p>Creates a static ARP cache entry associating the specified 32-bit IP address with the specified 48-bit hardware address.</p> <p>Note If an alias entry is created, then any interface to which the entry is attached will act as if it is the owner of the specified addresses, that is, it will respond to ARP request packets for this network layer address with the data link layer address in the entry.</p>
Step 3	commit	

Enabling Proxy ARP

Cisco IOS XR software uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the media addresses of hosts on other networks or subnets. For example, if the router receives an ARP request for a host that is not on the same interface as the ARP request sender, and if the router has all of its routes to that host through other interfaces, then it generates a proxy ARP reply packet giving its own local data-link address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host. Proxy ARP is disabled by default; this task describes how to enable proxy ARP if it has been disabled.

SUMMARY STEPS

1. **configure**
2. **interface *type number***
3. **proxy-arp**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface <i>type number</i> Example: RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/CPU0/0	Enters interface configuration mode.
Step 3	proxy-arp Example: RP/0/RP0/CPU0:router(config-if)# proxy-arp	Enables proxy ARP on the interface.
Step 4	commit	

Configure Learning of Local ARP Entries

You can configure an interface or a sub-interface to learn only the ARP entries from its local subnet.

Use the following procedure to configure local ARP learning on an interface.

- 1 Enter the interface configuration mode.

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1
```

Configure Learning of Local ARP Entries

- 2** Configure the IPv4/IPv6 address for the interface.

```
RP/0/RP0/CPU0:router (config-if) # ipv4 address 12.1.3.4 255.255.255.0
```

- 3** Configure local ARP learning on the interface.

```
RP/0/RP0/CPU0:router(config-if) # arp learning local
```

- 4** Enable the interface and commit your configuration.

```
RP/0/RP0/CPU0:router(config-if) # no shut
RP/0/RP0/CPU0:router(config-if) # commit
RP/0/0/CPU0:Dec 12 13:41:16.580 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/1, changed state to Down
RP/0/0/CPU0:Dec 12 13:41:16.683 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/1, changed state to Up
```

- 5** Confirm your configuration.

```
RP/0/RP0/CPU0:router(config-if) # show running-configuration
..
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Mon Dec 12 13:41:16 2016
!
interface GigabitEthernet0/0/0/1
  ipv4 address 12.1.3.4 255.255.255.0
  arp learning local
!
```

- 6** Verify if local ARP learning is working as configured on the interface.

```
RP/0/RP0/CPU0:router(config-if) # do show arp idb gigabitEthernet 0/0/0/1 location 0/0/CPU0
Thu Dec 15 10:00:11.733 IST

GigabitEthernet0/0/0/1 (0x00000040):
  IPv4 address 12.1.3.4, Vrf ID 0x60000000
  VRF Name default
  Dynamic learning: Local
  Dynamic entry timeout: 14400 secs
  Purge delay: off
  IPv4 caps added (state up)
  MPLS caps not added
  Interface not virtual, not client fwd ref,
  Proxy arp not configured, not enabled
  Local Proxy arp not configured
  Packet IO layer is NetIO
  Srg Role : DEFAULT
  Idb Flag : 2146444
  IDB is Complete
```

- 7** (Optional) You can monitor the ARP traffic on the interface.

```
RP/0/RP0/CPU0:router(config-if) # do show arp traffic gigabitEthernet 0/0/0/1 location 0/0/CPU0
Thu Dec 15 10:13:28.964 IST
```

```
ARP statistics:
  Recv: 0 requests, 0 replies
  Sent: 0 requests, 1 replies (0 proxy, 0 local proxy, 1 gratuitous)
  Subscriber Interface:
    0 requests recv, 0 replies sent, 0 gratuitous replies sent
  Resolve requests rcvd: 0
  Resolve requests dropped: 0
  Errors: 0 out of memory, 0 no buffers, 0 out of sunbet
```

```
ARP cache:
  Total ARP entries in cache: 1
  Dynamic: 0, Interface: 1, Standby: 0
  Alias: 0, Static: 0, DHCP: 0
  IP Packet drop count for GigabitEthernet0_0_0_1: 0
```