



# Implementing 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). This module describes how to configure ARP processes on the Cisco ASR 9000 Series Aggregation Services Router.



**Note** For a complete description of the ARP commands listed in this module, refer to the *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.

## Feature History for Configuring ARP

Release	Modification
Release 3.7.2	This feature was introduced.

- [Prerequisites for Configuring ARP](#) , on page 1
- [Restrictions for Configuring ARP](#) , on page 1
- [Information About Configuring ARP](#) , on page 2
- [How to Configure ARP](#) , on page 5
- [Configuration Examples for ARP Configuration on Cisco IOS XR Software](#), on page 12
- [Additional References](#), on page 14

## 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.

- Due to a hardware limitation in the Ethernet SPA interfaces installed on all routers, when a packet contains a wrong destination address, the corresponding SPA drops the packet even if the ingress packet count is already incremented in the output of the **show interfaces** command.
- ARP throttling is not supported.



---

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

---

The following additional restrictions apply when configuring the Direct Attached Gateway Redundancy (DAGR) feature on Cisco ASR 9000 Series Routers:

- IPv6 is not supported.
- Ethernet bundles are not supported.
- Non-Ethernet interfaces are not supported.
- Hitless ARP Process Restart is not supported.
- Hitless RSP Failover is not supported.

## 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. Cisco IOS XR software also supports a form of ARP called local proxy ARP.

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.

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

- The target IP address in the ARP request, the IP address of the ARP source, and the IP address of the interface on which the ARP request is received are on the same Layer 3 network.
- The next hop for the target IP address is through the same interface as the request is received.

Typically, local proxy ARP is used to resolve MAC addresses to IP addresses in the same Layer 3 network such as, private VLANs that are Layer 2-separated. Local proxy ARP supports all types of interfaces supported by ARP and unnumbered interfaces.

## 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.

## Direct Attached Gateway Redundancy

Direct Attached Gateway Redundancy (DAGR) allows third-party redundancy schemes on connected devices to use gratuitous ARP as a failover signal, enabling the ARP process to advertise a new type of route in the Routing Information Base (RIB). These routes are distributed by Open Shortest Path First (OSPF).

Sometimes part of an IP network requires redundancy without routing protocols. A prime example is in the mobile environment, where devices such as base station controllers and multimedia gateways are deployed in redundant pairs, with aggressive failover requirements (subsecond or less), but typically do not have the capability to use native Layer 3 protocols such as OSPF or Intermediate System-to-Intermediate System (IS-IS) protocol to manage this redundancy. Instead, these devices assume they are connected to adjacent IP devices over an Ethernet switch, and manage their redundancy at Layer 2, using proprietary mechanisms similar to Virtual Router Redundancy Protocol (VRRP). This requires a resilient Ethernet switching capability, and depends on mechanisms such as MAC learning and MAC flooding.

DAGR is a feature that enables many of these devices to connect directly to Cisco ASR 9000 Series Routers without an intervening Ethernet switch. DAGR enables the subsecond failover requirements to be met using a Layer 3 solution. No MAC learning, flooding, or switching is required.



---

**Note** Since mobile devices' 1:1 Layer 2 redundancy mechanisms are proprietary, they do not necessarily conform to any standard. So although most IP mobile equipment is compatible with DAGR, interoperability does require qualification, due to the possibly proprietary nature of the Layer 2 mechanisms with which DAGR interfaces.

---

## Additional Guidelines

The following are additional guidelines to consider when configuring DAGR:

- Up to 40 DAGR peers, which may be on the same or different interfaces, are supported per system.
- Failover is supported for DAGR routes within 500 ms of receipt of an ARP reply packet.
- On ARP process restart, DAGR groups are reinitialized.

## 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** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type*
  - **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* **alias**
3. **commit**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>	
Step 2	Do one of the following: <ul style="list-style-type: none"> <li>• <b>arp</b> [<b>vrf</b> <i>vrf-name</i>] <i>ip-address hardware-address encapsulation-type</i></li> <li>• <b>arp</b> [<b>vrf</b> <i>vrf-name</i>] <i>ip-address hardware-address encapsulation-type</i> <b>alias</b></li> </ul> <b>Example:</b> <pre>RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa</pre> or	Creates a static ARP cache entry associating the specified 32-bit IP address with the specified 48-bit hardware address.  <b>Note</b> If an <b>alias</b> 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.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa alias	
<b>Step 3</b>	<b>commit</b>	

## 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
<b>Step 1</b>	<b>configure</b>	
<b>Step 2</b>	<b>interface</b> <i>type number</i>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0	Enters interface configuration mode.
<b>Step 3</b>	<b>proxy-arp</b>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config-if)# proxy-arp	Enables proxy ARP on the interface.
<b>Step 4</b>	<b>commit</b>	

## Enabling Local Proxy ARP

Local proxy ARP is disabled by default; this task describes how to enable local proxy ARP.

### SUMMARY STEPS

1. **configure**

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

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>	
Step 2	<b>interface</b> <i>type number</i>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/0	Enters interface configuration mode.
Step 3	<b>local-proxy-arp</b>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config-if)# local-proxy-arp	Enables local proxy ARP on the interface.
Step 4	<b>commit</b>	

## Configuring DAGR

Follow these steps to create a DAGR group on the Cisco ASR 9000 Series Router.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **arp dagr**
4. **peer ipv4** *address*
5. **route distance normal** *normal- distance* **priority** *priority-distance*
6. **route metric normal** *normal- metric* **priority** *priority-metric*
7. **timers query** *query-time* **standby** *standby-time*
8. **priority-timeout** *time*
9. Do one of the following:
  - **end**
  - **commit**
10. **show arp dagr** [ *interface* [ *IP-address* ]]

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>  <b>Example:</b>	Enters global configuration mode.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router# configure	
<b>Step 2</b>	<p><b>interface</b> <i>type interface-path-id</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/0</pre>	Enters interface configuration mode and configures an interface.
<b>Step 3</b>	<p>arp dagr</p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if)# arp dagr</pre>	Enters DAGR configuration mode.
<b>Step 4</b>	<p><b>peer</b> <b>ipv4</b> <i>address</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr)# peer ipv4 10.0.0.100</pre>	Creates a new DAGR group for the virtual IP address.
<b>Step 5</b>	<p><b>route distance normal</b> <i>normal- distance</i> <b>priority</b> <i>priority-distance</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route distance normal 140 priority 3</pre>	(Optional) Configures route distance for the DAGR group.
<b>Step 6</b>	<p><b>route metric normal</b> <i>normal- metric</i> <b>priority</b> <i>priority-metric</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route metric normal 84 priority 80</pre>	(Optional) Configures the route metric for the DAGR group.
<b>Step 7</b>	<p><b>timers query</b> <i>query-time</i> <b>standby</b> <i>standby-time</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr-peer)# timers query 2 standby 19</pre>	(Optional) Configures the time in seconds between successive ARP requests being sent out for the virtual IP address.
<b>Step 8</b>	<p><b>priority-timeout</b> <i>time</i></p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr-peer)# priority-timeout 25</pre>	(Optional) Configures a timer for the length of time in seconds to wait before reverting to normal priority from a high-priority DAGR route.
<b>Step 9</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>• end</li> <li>• commit</li> </ul>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> <li>• When you issue the <b>end</b> command, the system prompts you to commit changes:</li> </ul>



	Command or Action	Purpose
	<p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr)# end</pre> <p>OR</p> <pre>RP/0/RSP0/CPU0:router(config-if-dagr)# commit</pre>	<p>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</p> <ul style="list-style-type: none"> <li>• Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>• Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>• Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> </ul> <p>Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</p>
<b>Step 10</b>	<p><b>show arp dagr</b> [ <i>interface</i> [ <i>IP-address</i> ]]</p> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router# show arp dagr</pre>	<p>(Optional) Displays the operational state of all DAGR groups. Using the optional <i>interface</i> and <i>IP-address</i> arguments restricts the output to a specific interface or virtual IP address.</p>

## Configuring ARP purge-delay

With Equal Cost Multi Path (ECMP), traffic is load balanced across multiple paths with equal cost. This should provide resiliency against interface flaps. If an interface goes down, the traffic is then routed via the other interface without traffic loss. However, if the first interface comes up, traffic is routed back over it but forwarding will only resume once ARP has been (re)resolved and the adjacency (re)installed. Here a short unexpected interface flap causes this traffic loss and is particularly undesirable.

The purge-delay feature allows existing dynamic entries to persist rather than immediately delete entries which could cause traffic loss following an interface flap.

The purge delay feature works by caching existing dynamic ARP entries when an interface goes down and starting a purge delay timer. When the interface is brought back and the purge delay timer not yet fired, the entries are reinstalled as before. The normal entry timeout is reduced in order to re-ARP for the entries after any interface state change related churn has died down; should the purge delay timer fire before the interface comes back up, the entries are deleted from the cache.

### SUMMARY STEPS

1. **configure**
- 2.
- 3.
4. **commit**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	<b>configure</b>	
<b>Step 2</b>	<b>Example:</b>  RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/	Enters interface configuration mode.
<b>Step 3</b>	<b>Example:</b>  RP/0/RSP0/CPU0:router(config-if)# arp purge-delay 100	Sets the purge delay time interval.
<b>Step 4</b>	<b>commit</b>	

# Configuring ARP timeout

Dynamic ARP entries which are learnt by ARP address resolution (when valid ARP replies are received) are timed out every 4 hours by default in order to remove stale entries.

ARP entries that correspond to the local interface or that are statically configured by the user never time out.

**SUMMARY STEPS**

- 1.
- 2.
- 3.
4. Do one of the following:
  - end
  - commit

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	<b>Example:</b>  RP/0/RSP0/CPU0:router# configure	Enters global configuration mode.
<b>Step 2</b>	<b>Example:</b>  RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/	Enters interface configuration mode.
<b>Step 3</b>	<b>Example:</b>  RP/0/RSP0/CPU0:router(config-if)# arp timeout 100	Sets the ARP cache timeout interval.
<b>Step 4</b>	Do one of the following:	Saves configuration changes.

	Command or Action	Purpose
	<ul style="list-style-type: none"> <li>• end</li> <li>• commit</li> </ul> <p><b>Example:</b></p> <pre>RP/0/RSP0/CPU0:router(config-if)# end</pre> <p>or</p> <pre>RP/0/RSP0/CPU0:router(config-if)# commit</pre>	<ul style="list-style-type: none"> <li>• When you issue the <b>end</b> command, the system prompts you to commit changes: <ul style="list-style-type: none"> <li>Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:</li> </ul> </li> <li>• Entering <b>yes</b> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</li> <li>• Entering <b>no</b> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</li> <li>• Entering <b>cancel</b> leaves the router in the current configuration session without exiting or committing the configuration changes.</li> <li>• Use the <b>commit</b> command to save the configuration changes to the running configuration file and remain within the configuration session.</li> </ul>

## 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.

```
Router(config)# interface GigabitEthernet 0/0/0/1
```

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

```
Router(config-if)# ipv4 address 12.1.3.4 255.255.255.0
```

3. Configure local ARP learning on the interface.

```
Router(config-if)# arp learning local
```

4. Enable the interface and commit your configuration.

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

5. Confirm your configuration.

```
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 GigabitEthernet 0/0/0/1
  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.

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

```
GigabitEthernet 0/0/0/1 (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.

```
Router(config-if)# do show arp idb 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 rcvd, 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
```

## Configuration Examples for ARP Configuration on Cisco IOS XR Software

This section provides the following ARP configuration examples:

### Creating a Static ARP Cache Entry: Example

The following is an example of a static ARP entry for a typical Ethernet host:

```
configure
arp 192.168.7.19 0800.0900.1834 arpa
```

The following is an example of a static ARP entry for a typical Ethernet host where the software responds to ARP requests as if it were the owner of both the specified IP address and hardware address, whether proxy ARP is enabled or not:

```
configure
arp 192.168.7.19 0800.0900.1834 arpa alias
```

The following is an example of configuring a static arp entry on an SRP device:

```
configure
arp 192.168.8.20 0800.0900.1723 srp
```

## Enabling Proxy ARP: Example

The following is an example of enabling proxy ARP:

```
configure
interface MgmtEth 0/
RSP0

/CPU0/0
proxy-arp
```

## Displaying the ARP Table: Example

The following example shows how to display the ARP table:

```
Router# show arp

-----
0/4/CPU0
-----
Address      Age      Hardware Addr  State   Type   Interface
-----
10.0.0.7     -        0014.a912.b7a0 Interface  ARPA   GigabitEthernet0/2/0/0
10.0.0.200   -        0001.0002.0003 Static    ARPA   GigabitEthernet0/2/0/0
10.0.0.201   -        0004.0005.0006 Alias     ARPA   GigabitEthernet0/2/0/0
10.0.0.252   00:28:20 0015.171f.9ef5 Dynamic  ARP!   GigabitEthernet0/2/0/0
```

## Enabling DAGR and Configuring a DAGR Group: Example

The following is an example of enabling DAGR and configuring a DAGR group peer:

```
configure
interface gigabitethernet 0/1/0/0.1
arp dagr
peer ipv4 192.168.7.19
priority-timeout 25
route distance normal 48 priority 5
route metric normal 48 priority 5
timers query 2 standby 40
commit
```

## Displaying the Operational State of DAGR Groups: Example

The following example shows how to display the current operational state of the DAGR groups:

```
RP/0/RSP0/CPU0:router# show arp dagr
-----
0/1/CPU0
-----
Interface                Virtual IP      State   Query-pd  Dist  Metr
GigabitEthernet0/1/0/2   10.168.7.19    Active None      150   100
GigabitEthernet0/1/0/2   10.24.0.45     Query  1         None  None
GigabitEtherget0/1/0/3   10.66.0.45     Init   None      None  None
```

## Additional References

The following sections provide references related to ARP.

### Related Documents

Related Topic	Document Title
ARP commands	<i>ARP Commands</i> module in <i>IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers</i>
Getting started material	<i>Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide</i>

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Quality of Service Commands</i> module in <i>Modular Quality of Service Command Reference for Cisco ASR 9000 Series Routers</i>
Class-based traffic shaping, traffic policing, low latency queuing, and MDDR	Configuring Modular Quality of Service Congestion Management module in <i>Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers</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.	—

### MIBs

MIBs	MIBs Link
—	To locate and download MIBs, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index">https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index</a>

**RFCs**

<b>RFCs</b>	<b>Title</b>
RFC 826	Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware
RFC 1027	Using ARP to implement transparent subnet gateways

**Technical Assistance**

<b>Description</b>	<b>Link</b>
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	<a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a>

