

Address Resolution Protocol

Last Updated: January 30, 2013

The Address Resolution Protocol (ARP) feature performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco systems that run IP.

This feature module explains ARP for IP routing and the optional ARP features you can configure, such as static ARP entries, timeout for dynamic ARP entries, clearing the cache, and proxy ARP.

- Finding Feature Information, page 1
- Information About the Address Resolution Protocol, page 1
- How to Configure the Address Resolution Protocol, page 6
- Configuration Examples for the Address Resolution Protocol, page 14
- Additional References, page 15
- Feature Information for the Address Resolution Protocol, page 16

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About the Address Resolution Protocol

- Layer 2 and Layer 3 Addressing, page 2
- Overview of the Address Resolution Protocol, page 3
- ARP Caching, page 4
- Static and Dynamic Entries in the ARP Cache, page 4
- Devices That Do Not Use ARP, page 4

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Inverse ARP, page 4
- Reverse ARP, page 5
- Proxy ARP, page 5
- Serial Line Address Resolution Protocol, page 6

Layer 2 and Layer 3 Addressing

IP addressing occurs at Layer 2 (data link) and Layer 3 (network) of the Open System Interconnection (OSI) reference model. OSI is an architectural network model developed by ISO and ITU-T that consists of seven layers, each of which specifies particular network functions such as addressing, flow control, error control, encapsulation, and reliable message transfer.

Layer 2 addresses are used for local transmissions between devices that are directly connected. Layer 3 addresses are used for indirectly connected devices in an internetwork environment. Each network uses addressing to identify and group devices so that transmissions can be sent and received. Ethernet (802.2, 802.3, Ethernet II, and Subnetwork Access Protocol [SNAP]), Token Ring, and Fiber Distributed Data Interface (FDDI) use media access control (MAC) addresses that are “burned in” to the network interface card (NIC). The most commonly used network types are Ethernet II and SNAP.

Note For the supported interface types, see the data sheet for your hardware platform.

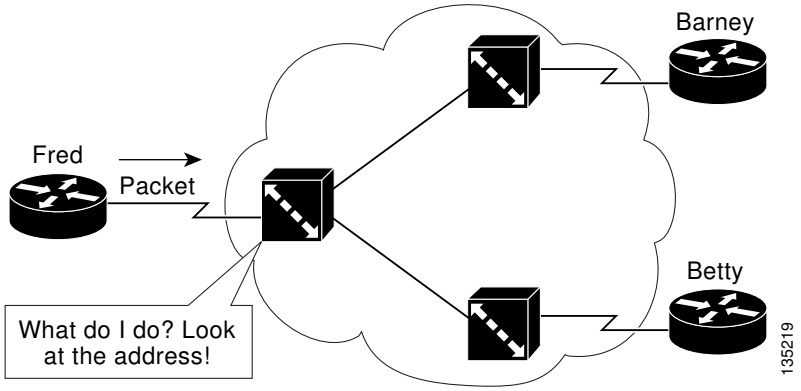
In order for devices to be able to communicate with each when they are not part of the same network, the 48-bit MAC address must be mapped to an IP address. Some of the Layer 3 protocols used to perform the mapping are:

- Address Resolution Protocol (ARP)
- Reverse ARP (RARP)
- Serial Line ARP (SLARP)
- Inverse ARP

For the purposes of IP mapping, Ethernet, Token Ring, and FDDI frames contain the destination and source addresses. Frame Relay and Asynchronous Transfer Mode (ATM) networks, which are packet-switched, data packets take different routes to reach the same destination. At the receiving end, the packet is reassembled in the correct order.

In a Frame Relay network, there is one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI), which identifies each VC. For example, in the figure below, the Frame Relay switch to which device Fred is connected receives frames; the switch forwards the frames to either Barney or Betty based on the DLCI that identifies each VC. So Fred has one physical connection but multiple logical connections.

Figure 1 Frame Relay Network



ATM networks use point-to-point serial links with the High-Level Data Link Control (HDLC) protocol. HDLC includes a meaningless address field included in five bytes of the frame header frame with the recipient implied since there can be only one.

Overview of the Address Resolution Protocol

The Address Resolution Protocol (ARP) was developed to enable communications on an internetwork and is defined by RFC 826. Layer 3 devices need ARP to map IP network addresses to MAC hardware addresses so that IP packets can be sent across networks. Before a device sends a datagram to another device, it looks in its ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network. Each device compares the IP address to its own. Only the device with the matching IP address replies to the sending device with a packet containing the MAC address for the device (except in the case of "proxy ARP"). 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. The figure below illustrates the ARP broadcast and response process.

Figure 2 ARP Process



When the destination device lies on a remote network, one beyond another Layer 3 device, the process is the same except that the sending device 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 Layer 3 device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet.

Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).

The ARP request message has the following fields:

- HLN—Hardware address length. Specifies how long the hardware addresses are in the message. For IEEE 802 MAC addresses (Ethernet) the value is 6.
- PLN—Protocol address length. Specifies how long the protocol (Layer 3) addresses are in the message. For IPv4, the value is 4.
- OP—Opcode. Specifies the nature of the message by code:
  - 1—ARP request.
  - 2—ARP reply.
  - 3 through 9—RARP and Inverse ARP requests and replies.
- SHA—Sender hardware address. Specifies the Layer 2 hardware address of the device sending the message.
- SPA—Sender protocol address. Specifies the IP address of the sending device.
- THA—Target hardware address. Specifies the Layer 2 hardware address of the receiving device.
- TPA—Target protocol address. Specifies the IP address of the receiving device.

ARP Caching

Because the mapping of IP addresses to media access control (MAC) addresses occurs at each hop (Layer 3 device) on the network for every datagram sent over an internetwork, performance of the network could be compromised. To minimize broadcasts and limit wasteful use of network resources, Address Resolution Protocol (ARP) caching was implemented.

ARP caching is the method of storing network addresses and the associated data-link addresses in memory for a period of time as the addresses are learned. This minimizes the use of valuable network resources to broadcast for the same address each time a datagram is sent. The cache entries must be maintained because the information could become outdated, so it is critical that the cache entries are set to expire periodically. Every device on a network updates its tables as addresses are broadcast.

There are static ARP cache entries and dynamic ARP cache entries. Static entries are manually configured and kept in the cache table on a permanent basis. Static entries are best for devices that have to communicate with other devices usually in the same network on a regular basis. Dynamic entries are added by Cisco software, kept for a period of time, and then removed.

Static and Dynamic Entries in the ARP Cache

Static routing requires an administrator to manually enter into a table IP addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other. The table is built and changed automatically. No administrative tasks are needed unless a time limit is added, so dynamic routing is more efficient than static routing. The default time limit is 4 hours. If the network has many routes that are added and deleted from the cache, the time limit should be adjusted.

The routing protocols that dynamic routing uses to learn routes, such as distance-vector and link-state routing protocols, are beyond the scope of this document.

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 Media Access Control (MAC) addresses. The bridge builds its own address table, which uses MAC addresses only, as opposed to a router, which has an Address Resolution Protocol (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 all ports to the devices and operate at Layer 1, but they 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 the message only to that port, unlike a hub, which sends the message out all its ports. However, Layer 3 switches are routers that build an ARP cache (table).

Inverse ARP

Inverse ARP, which is enabled by default in ATM networks, builds an ATM map entry and is necessary to send unicast packets to a server (or relay agent) on the other end of a connection. Inverse ARP is supported only for the **aal5snap** encapsulation type.

For multipoint interfaces, an IP address can be acquired using other encapsulation types because broadcast packets are used. However, unicast packets to the other end will fail because there is no ATM map entry and thus DHCP renewals and releases also fail.

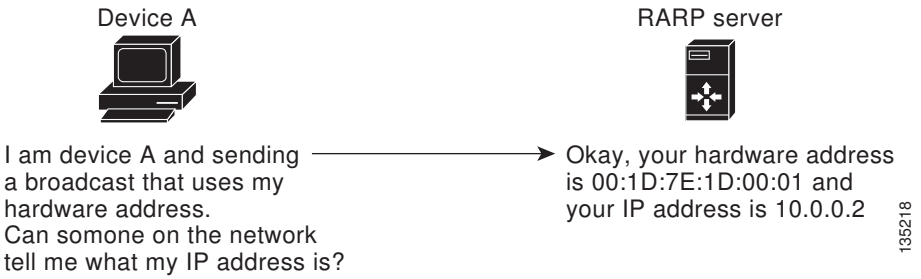
For more information about Inverse ARP and ATM networks, see the "Configuring ATM" feature module in the *Asynchronous Transfer Mode Configuration Guide*.

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as the Address Resolution Protocol (ARP), except that the RARP request packet requests an IP address instead of a media access control (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 in to the hardware.

RARP requires a RARP server on the same network segment as the device interface. The figure below illustrates how RARP works.

Figure 3 RARP Process



Because of the limitations with RARP, most businesses use Dynamic Host Configuration Protocol (DHCP) to assign IP addresses dynamically. DHCP is cost-effective and requires less maintenance than RARP. The most important limitations with RARP are as follows:

- 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 the IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts but not subnet masks or default gateways.

Cisco software attempts to use RARP if it does not know the IP address of an interface at startup to respond to RARP requests that it is able to answer. The AutoInstall feature of the software automates the configuration of Cisco devices.

AutoInstall supports RARP and enables a network manager to connect a new device to a network, turn it on, and automatically load a pre-existing configuration file. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, see the *Configuration Fundamentals Configuration Guide*.

Proxy ARP

Proxy Address Resolution Protocol, as defined in RFC 1027, was implemented to enable devices that are separated into physical network segments connected by a router in the same IP network or subnetwork to

resolve IP-to-MAC addresses. When devices are not in the same data link layer network but are in the same IP network, they try to transmit data to each other as if they were on the local network. However, the router that separates the devices will not send a broadcast message because routers do not pass hardware-layer broadcasts. Therefore, the addresses cannot be resolved.

Proxy ARP is enabled by default so the “proxy router” that resides between the local networks responds with its MAC address as if it were the router to which the broadcast is addressed. When the sending device receives the MAC address of the proxy router, it sends the datagram to the proxy router, which in turns sends the datagram to the designated device.

Proxy ARP is invoked by 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 proxy ARP is disabled, a device responds to ARP requests received on its interface only if the target IP address is the same as its IP address or if the target IP address in the ARP request has a statically configured ARP alias.

Serial Line Address Resolution Protocol

Serial Line ARP (SLARP) is used for serial interfaces that use High-Level Data Link Control (HDLC) encapsulation. A SLARP server, intermediate (staging) device, and another device providing a SLARP service might be required in addition to a TFTP server. If an interface is not directly connected to a server, the staging device is required to forward the address-resolution requests to the server. Otherwise, a directly connected device with SLARP service is required. Cisco software attempts to use SLARP if it does not know the IP address of an interface at startup to respond to SLARP requests that software is able to answer.

Cisco software automates the configuration of Cisco devices with the AutoInstall feature. AutoInstall supports SLARP and enables a network manager to connect a new device to a network, turn it on, and automatically load a pre-existing configuration file. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, see the *Configuration Fundamentals Configuration Guide*.

Note AutoInstall supports serial interfaces that use Frame Relay encapsulation.

How to Configure the Address Resolution Protocol

By default, the Address Resolution Protocol (ARP) feature is enabled and is set to use Ethernet encapsulation. Perform the following tasks to change or verify ARP functionality:

- Enabling the Interface Encapsulation, page 7
- Defining Static ARP Entries, page 8
- Setting an Expiration Time for Dynamic Entries in the ARP Cache, page 9
- Globally Disabling Proxy ARP, page 10
- Disabling Proxy ARP on an Interface, page 11
- Verifying the ARP Configuration, page 13

Enabling the Interface Encapsulation

Perform this task to support a type of encapsulation for a specific network, such as Ethernet, Frame Relay, FDDI, or Token Ring. When Frame Relay encapsulation is specified, the interface is configured for a Frame Relay subnetwork with one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI) that identifies each VC. When SNAP encapsulation is specified, the interface is configured for FDDI or Token Ring networks.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Defining Static ARP Entries" task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **arp** {**arpa** | **frame-relay** | **snap**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**arp** {**arpa** | **frame-relay** | **snap**} **Example:** `Device(config-if)# arp arpa`	Specifies the encapsulation type for an interface by type of network, such as Ethernet, FDDI, Frame Relay, and Token Ring. The keywords are as follows: • **arpa**—Enables encapsulation for an Ethernet 802.3 network. • **frame-relay**—Enables encapsulation for a Frame Relay network. • **snap**—Enables encapsulation for FDDI and Token Ring networks.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Defining Static ARP Entries

Perform this task to define static mapping between an IP address (32-bit address) and a Media Access Control (MAC) address (48-bit address) for hosts that do not support dynamic Address Resolution Protocol (ARP). Because most hosts support dynamic address resolution, defining static ARP cache entries is usually not required. Performing this task installs a permanent entry in the ARP cache that never times out. The entries remain in the ARP table until they are removed using the **no arp** command or the **clear arp interface** command for each interface.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Enabling the Interface Encapsulation" task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*] **Example:** `Device(config)# arp 10.0.0.0 aabb.cc03.8200 arpa`	Globally associates an IP address with a MAC address in the ARP cache. • *ip-address*—IP address in four-part dotted decimal format corresponding to the local data-link address. • **vrf** *vrf-name*—Virtual routing and forwarding instance for a Virtual Private Network (VPN). The *vrf-name* argument is the name of the VRF table. • *hardware-address*—Local data-link address (a 48-bit address). • *encap-type*—Encapsulation type for the static entry. The keywords are as follows: ◦ **arpa**—For Ethernet interfaces. ◦ **sap**—For Hewlett Packard interfaces. ◦ **smds**—For Switched Multimegabit Data Service (SMDS) interfaces. ◦ **snap**—For FDDI and Token Ring interfaces. ◦ **srp-a**—Switch route processor side A (SRP-A) interfaces. ◦ **srp-b**—Switch route processor side B (SRP-B) interfaces. **Note** Some keywords might not apply to your hardware platform. • *interface-type*—(Optional) Interface type (for more information, use the question mark (?) online help).
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Setting an Expiration Time for Dynamic Entries in the ARP Cache

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **arp timeout** *seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.
Step 4	**arp timeout** *seconds* **Example:** `Device(config-if)# arp timeout 30`	Sets the length of time, in seconds, an Address Resolution Protocol (ARP) cache entry stays in the cache. A value of zero means that entries are never cleared from the cache. The default is 14400 seconds (4 hours). **Note** If the network has frequent changes to cache entries, change the default to a shorter time period.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Globally Disabling Proxy ARP

Proxy Address Resolution Protocol (ARP) is enabled by default; perform this task to globally disable proxy ARP on all interfaces.

The Cisco software uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the media access control (MAC) addresses of hosts on other networks or subnets. For example, if hosts A and B are on different physical networks, host B does not receive the ARP broadcast request from host A and cannot respond to it. However, if the physical network of host A is connected by a gateway to the physical network of host B, the gateway sees the ARP request from host A.

Assuming that subnet numbers were assigned to correspond to physical networks, the gateway can also tell that the request is for a host that is on a different physical network. The gateway can then respond for host

B, saying that the network address for host B is that of the gateway itself. Host A sees this reply, caches it, and sends future IP packets for host B to the gateway.

The gateway forwards such packets to host B by using the configured IP routing protocols. The gateway is also referred to as a transparent subnet gateway or ARP subnet gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip arp proxy disable**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip arp proxy disable** **Example:** `Device(config)# ip arp proxy disable`	Disables proxy ARP on all interfaces. • The **ip arp proxy disable** command overrides any proxy ARP interface configuration. • To reenable proxy ARP, use the **no ip arp proxy disable** command. • You can also use the **default ip proxy arp** command to return to the default proxy ARP behavior, which is enabled.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Disabling Proxy ARP on an Interface

Proxy Address Resolution Protocol (ARP) is enabled by default; perform this task to disable proxy ARP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip proxy-arp**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.
Step 4	**no ip proxy-arp** **Example:** `Device(config-if)# no ip proxy-arp`	Disables proxy ARP on the interface. • To reenable proxy ARP, use the **ip proxy-arp** command. • You can also use the **default ip proxy-arp** command to return to the default proxy ARP behavior on the interface, which is enabled.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the ARP Configuration

SUMMARY STEPS

1. **show interfaces**
2. **show arp**
3. **show ip arp**
4. **show processes cpu | include (ARP | PID)**

DETAILED STEPS

Step 1 **show interfaces**

To display the type of Address Resolution Protocol (ARP) being used on a particular interface and also display the ARP timeout value, use the **show interfaces** privileged EXEC command.

Example:

```
Device# show interfaces GigabitEthernet0/0/0

GigabitEthernet0/0/0 is up, line protocol is up
 Hardware is SPA-8X1GE-V2, address is 001a.3045.4100 (bia 001a.3045.4100)
 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
 reliability 255/255, txload ½55, rxload ½55
 Encapsulation ARPA, loopback not set
 Keepalive not supported
 Full Duplex, 1000Mbps, link type is auto, media type is SX
 output flow-control is off, input flow-control is off
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output 00:00:50, output hang never
 Last clearing of ''show interface'' counters never
 Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/40 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
 0 packets input, 0 bytes, 0 no buffer
 Received 0 broadcasts (0 IP multicasts)
 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 0 watchdog, 0 multicast, 0 pause input
 7998 packets output, 3074275 bytes, 0 underruns
 0 output errors, 0 collisions, 4 interface resets
 0 babbles, 0 late collision, 0 deferred
 0 lost carrier, 0 no carrier, 0 pause output
 0 output buffer failures, 0 output buffers swapped out
```

Step 2 **show arp**

Use the **show arp** privileged EXEC command to examine the contents of the ARP cache.

Example:

```
Device# show arp

Protocol    Address       Age (min)   Hardware Addr      Type    Interface
Internet    10.1.1.1      43          001b.53e1.7201     ARPA    GigabitEthernet0/0/6
Internet    10.1.1.2      29          0021.d8ab.0b00     ARPA    GigabitEthernet0/0/6
```

```
Internet    10.1.2.1      80          001a.3045.4107     ARPA    GigabitEthernet0/0/7
Internet    10.1.2.1       -          0000.0c02.a03c     ARPA    GigabitEthernet0/0/7
```

Step 3 **show ip arp**

Use the **show ip arp** privileged EXEC command to show IP entries. To remove all nonstatic entries from the ARP cache, use the **clear arp-cache** privileged EXEC command.

Example:

```
Device# show ip arp

Protocol    Address       Age (min)   Hardware Addr      Type    Interface
Internet    10.1.1.1      43          001b.53e1.7201     ARPA    GigabitEthernet0/0/6
Internet    10.1.1.2      29          0021.d8ab.0b00     ARPA    GigabitEthernet0/0/6
Internet    10.1.2.1      80          001a.3045.4107     ARPA    GigabitEthernet0/0/7
Internet    10.1.2.1       -          0000.0c02.a03c     ARPA    GigabitEthernet0/0/7
```

Step 4 **show processes cpu** | **include** (**ARP** | **PID**)

Use the **show processes cpu** | **include** (**ARP** | **PID**) command to display ARP and RARP processes.

Example:

```
Device# show processes cpu | include (ARP | PID)

PID  Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   9          46       515         89  0.00%  0.00%  0.00%   0  ARP Input
  10           7     19078          0  0.00%  0.00%  0.00%   0  ARP Background
 110           1         2        500  0.00%  0.00%  0.00%   0  IP ARP Adjacency
 136           0         7          0  0.00%  0.00%  0.00%   0  ARP HA
 182           0         8          0  0.00%  0.00%  0.00%   0  RARP Input
```

Configuration Examples for the Address Resolution Protocol

- Example: Static ARP Entry Configuration, page 14
- Example: Encapsulation Type Configuration, page 15
- Example: Proxy ARP Configuration, page 15

Example: Static ARP Entry Configuration

The following example shows how to configure a static Address Resolution Protocol (ARP) entry in the cache by using the **alias** keyword, allowing the software to respond to ARP requests as if it were the interface of the specified address:

```
arp 10.0.0.0 aabb.cc03.8200 alias
interface gigabitethernet0/0/0
```

Example: Encapsulation Type Configuration

The following example shows how to configure the encapsulation on the interface. The **arpa** keyword indicates that interface is connected to an Ethernet 802.3 network:

```
interface gigabitethernet0/0/0
 ip address 10.108.10.1 255.255.255.0
 arp arpa
```

Example: Proxy ARP Configuration

The following example shows how to configure proxy ARP because it was disabled for the interface:

```
interface gigabitethernet0/0/0
 ip proxy-arp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
ARP commands	*Cisco IOS IP Addressing Services Command Reference*
AppleTalk addressing scheme	Core Competence AppleTalk (white paper) at www.corecom.com/html/appletalk.html
Authorized ARP	"Configuring DHCP Services for Accounting and Security" feature module in the *IP Addressing: DHCP Configuration Guide* (part of the *IP Addressing Configuration Guide Library*)
Inverse ARP and ATM networks	"Configuring ATM" feature module in the *Asynchronous Transfer Mode Configuration Guide*
AutoInstall	*Configuration Fundamentals Configuration Guide*

RFCs

RFCs	Title
RFC 826	*Address Resolution Protocol*
RFC 903	*Reverse Address Resolution Protocol*
RFC 1027	*Proxy Address Resolution Protocol*

RFCs	Title
RFC 1042	*Standard for the Transmission of IP Datagrams over IEEE 802 Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the Address Resolution Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for the Address Resolution Protocol

Feature Name	Software Releases	Feature Information
Address Resolution Protocol	12.2(15)T 15.0(1)S Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2SE	The Address Resolution Protocol (ARP) feature performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco systems that run IP.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks.

Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2013 Cisco Systems, Inc. All rights reserved.