

Chapter 1

Routing Apollo Domain

1

This chapter describes how to configure Cisco Systems' implementation of the Apollo Domain routing protocol. Tasks and topics described in this chapter include:

- An overview of Apollo Domain
- How to enable Apollo Domain routing
- How to configure static routes, set update timers, and define access lists

Global and interface command summaries appear at the end of the chapter.

Cisco's Implementation of Apollo Domain

The Apollo Domain routing protocol is the native-mode networking protocol for Apollo workstations. The Cisco routing software implementation supports packet forwarding and routing for the Apollo Domain network protocols on Ethernet, FDDI and serial interfaces using HDLC or X.25 encapsulation. Direct attachment to the 12-megabit Domain Token Ring is not supported.

The following restrictions apply to the Cisco implementation:

- If bridging is enabled on an Ethernet for which Apollo Domain routing is also enabled, special care must be taken. An Ethernet type code access list must be specified that filters out datagrams with the Apollo Domain type code (hexadecimal 8019). This restriction applies to MCI cards running microcode Version 1.5 or earlier.
- An IP address must be set on all media that use the ARP protocol (Ethernet and FDDI, for example). This is because Domain ARP uses the same Ethernet type value as IP ARP.
- The Cisco implementation of the Apollo Domain routing support assumes that it can use ARP to locate workstations on the local cable. Following are the versions of the Apollo operating system that support Domain ARP (D-ARP):
 - DN3000 and DN3010 nodes need Version 9.7.4.1. This version of the operating system is available on a patch (ask for patch 186) from local Apollo field offices.
 - DN3500, 4000, and 4500 nodes need Version 9.7.5.1 which is available on patch tape 185.
 - Version 9.7 (which provides ARP) for DN5xx-T nodes need Version 9.7.4.b101. No patch is available for these machines; it is provided only on a DECnet tape.

Note: Version 10.0 does not provide ARP. You must migrate to Version 10.1 and versions of the Apollo Domain operating system before successfully operating with Cisco routers. The **rtchk** and **lcnod**e commands and D-ARP are not supported in Apollo's 802.5 implementation by Cisco routers.

Apollo Domain Addresses

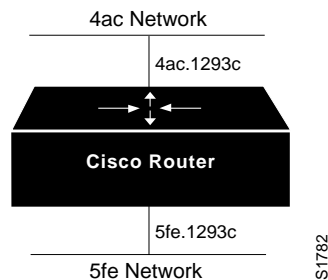
Apollo Domain addresses are 20-bit quantities, represented by five-digit hexadecimal numbers. Each host has a single address which is used for all of its network connections.

An Apollo Domain host may have interfaces on more than one physical network (Ethernet, Domain Token Ring, serial line, and so on). Physical networks are identified by 32-bit numbers written in hexadecimal. These network numbers must be unique throughout an Apollo Domain internet. As both the network number and the host address are needed to deliver traffic to a host, addresses are usually given as network numbers followed by host addresses separated with dots. An example would be:

5fe.1293c

The number 5fe identifies the physical network, and the number 1293c identifies the host as shown in Figure 1-1.

Figure 1-1 Apollo Domain Addresses



Configuring Apollo Domain Routing

There are only two commands required to enable Apollo Domain routing:

Step 1: Use the global configuration command **apollo routing** to enable routing.

Step 2: Use the interface subcommand **apollo network** to assign apollo routing to a specific interface.

All other configuration commands provide additional functionality or refinements. Each task is described in the following section. These descriptions are followed by applicable EXEC commands for monitoring and debugging Apollo Domain networks. Summaries of global configuration commands and interface subcommands described here appear at the end of this chapter.

Enabling Apollo Domain Routing

To enable or disable Domain routing and specify which system-wide host address to use, use the **apollo routing** global configuration command. The full syntax of this command follows.

```
apollo routing address  
no apollo routing
```

The argument *address* is a unique, five-digit hexadecimal host address that you define, as described in the section “Apollo Domain Addresses.” The **no apollo routing** command disables Apollo routing. See the next section for an example of how to enable Apollo Domain routing.

Assigning the Apollo Domain Network Numbers

Apollo Domain network numbers must be assigned to the appropriate interfaces. This is done using the **apollo network** interface subcommand. The full syntax of this command follows.

```
apollo network number  
no apollo network
```

The argument *number* is an eight-digit hexadecimal number. The **no apollo network** command removes a network number from an interface.

Example:

The following is a very simple example of setting up Apollo Domain routing on a router with two Ethernet interfaces. The first step is to enable the RIP routing protocol and assign a Domain network address using the **apollo routing** command. The next step is to assign network numbers to the two interfaces.

```
apollo routing 23d5a
interface ethernet 0
apollo network 5f
interface ethernet 1
apollo network 4e
```

Configuring Static Routes

Specify static routes for an Apollo Domain network with the **apollo route** global configuration command. Full syntax of this command follows:

```
apollo route network network.address
no apollo route network network.address
```

Use of this command causes packets received for the specified network to be forwarded to the specified router (whose address is *network.address*), whether or not that router is sending out dynamic routing. Use the **no apollo route** command to remove the routes.

Example:

If the router that handled traffic for network 33 had the address 45.91ac6, you would enter the following command.

```
apollo route 33 45.91ac6
```

Configuring Maximum Paths

To set the maximum number of multiple paths that the router will remember and use, use the **apollo maximum-paths** global configuration command. The command increases throughput by using multiple paths. It remembers higher bandwidth routes in preference to lower bandwidth routes. Full syntax of this command follows:

```
apollo maximum-paths paths
no apollo maximum-paths
```

The argument *paths* is the number of paths to be assigned. For a given destination, multiple paths of equal cost will be remembered. Output will be determined in round-robin fashion over these multiple paths at the packet level. The default value for *paths* is one; the **no apollo maximum-paths** command restores this default.

Example:

The following command sets three maximum paths.

```
apollo maximum-paths 3
```

The EXEC command **show apollo route** displays these additional routes and the maximum path value.

Setting Apollo Update Timers

To allow the Apollo Domain routing update timers to be set on a per-interface basis, use the **apollo update-time** interface subcommand:

```
apollo update-time seconds
```

Internal Apollo Domain routing timers are affected by the value set for the *seconds* argument, as follows:

- Apollo Domain routes are marked invalid if no routing updates are heard within six times the value of the update timer.
- Apollo Domain routes are removed from the routing table if no routing updates are heard within eight times the value of the update timer.
- The default value for the *seconds* argument is 30.
- The minimum value of the *seconds* argument is 10 seconds.
- The granularity of the update timer is determined by the lowest value defined.

Example:

In the example below, the granularity of the update timer is 20 because that is the lowest value specified.

```
interface serial 0
apollo update-time 40
interface ethernet 0
apollo update-time 20
interface ethernet 1
apollo update-time 25
```

Note: Only use this command in an all-Cisco environment, and ensure that all timers are the same for all routers attached to the same network segment.

The EXEC command **show apollo interface** displays the value of these timers.

Configuring Apollo Domain Access Lists

Apollo Domain access lists are a collection of permit and deny conditions that apply to defined Apollo network and host numbers. The router sequentially tests the network and host numbers against conditions set in the access lists.

The first match determines whether the router accepts or rejects the network and host number. Because the router stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the network and host number.

Specifying Apollo Domain Access Lists

Use the **apollo access-list** global configuration command to specify an access condition. The full syntax of this command follows:

```
apollo access-list name {permit | deny} [firstnet]lastnet.host [wildcard-mask]  
no apollo access-list name
```

The argument *name* is a name defined by the network administrator for the access list. The **no apollo access-list** command removes an access list; use only the name, not all the possible parameters, when you remove the list.

Choose the permit or deny condition for this list using the **permit** or **deny** keyword.

You may define Apollo access lists for one or a selected range of Apollo networks, which are defined by network number and host number separated by a dot. The optional argument *firstnet* and the argument *lastnet.host* specify a selected network range. Use the argument *lastnet.host* to specify just one network. Use -1 as a net number to specify all networks.

The optional *wildcard-mask* argument is a wildcard mask that uses the one bits to ignore the host part of the network address. Host bits corresponding to wildcard mask bits set to zero are used in comparisons.

An access list can contain an indefinite number of actual and wildcard addresses. A wildcard address has a nonzero mask and thus potentially matches more than one actual address. The software examines the actual addresses, then the wildcard addresses. The order of the wildcard addresses is important because the software stops examining access list entries once it finds a match.

Protocol types and/or socket numbers are not useful in Apollo access lists. Also, note that Apollo access lists are named, not numbered as they are with other protocols.

Use the **no apollo access-list** command to delete the entire access list.

Example:

In the example below, the first line denies access to networks 3a through 3f, the second line denies access to a specific host and the third line permits everyone else.

```
apollo access-list eng deny 3a-3f.0 fffff
apollo access-list eng deny 5fe.1293c
apollo access-list eng permit -1.0 ffff
```

Defining Access Groups

Use the **apollo access-group** interface subcommand to specify the interface on which the access list is defined. Full syntax of this command follows:

```
apollo access-group name
no apollo access-group name
```

Enter the user-defined *name* for the access list defined by the **apollo access-list** global configuration command for the argument *name*.

Upon receiving and routing a packet to a controlled interface, the software checks the source network and host number of the packet against that set in the access list. If the access list permits the address, the software transmits the packet.

You can specify ranges of network numbers, along with host masks. While the masks may not be useful, they permit the host part to be ignored entirely.

Use the **no apollo access-group** command to remove the name.

Example:

In the example below, the access list named eng is assigned to the first Ethernet interface.

```
interface ethernet 0
apollo access-group eng
```

Monitoring the Apollo Domain Network

Use the EXEC commands described in this section to obtain displays of activity on the Apollo Domain network.

Displaying Apollo Interface Parameters

Use the EXEC command **show apollo interface** to display Apollo Domain parameters that have been configured on the interfaces. Enter this command at the EXEC prompt:

```
show apollo interface [interface unit]
```

You may specify the optional *interface* and *unit* arguments to see information for just that interface.

Following is sample output, specifying the first Ethernet interface:

```
Ethernet 0 is up, line protocol is up
  Apollo address is 123A.CAFE
  Update time is 30 seconds
  Outgoing access list is not set
```

Displaying Apollo Routes

Use the EXEC command **show apollo route** to display the Apollo Domain routing table. Enter this command at the EXEC prompt:

```
show apollo route
```

Following is sample output:

```
Codes: R - RIP derived, C - connected, S - static, 1 learned routes

Maximum allowed path(s) are/is 1
C Net 123A is directly connected, 0 uses, Ethernet0
C Net 123B is directly connected, 0 uses, Ethernet1
R Net 123C [1/0] via 123A.CAFB, 4 sec, 0 uses, Ethernet0
```

In the display, the leading character R indicates routes learned via RIP, C indicates connected entries, and S indicates statically defined entries.

Displaying Apollo Traffic Statistics

Use the EXEC command **show apollo traffic** to display information on the number and type of Apollo Domain packets transmitted and received. Enter this command at the EXEC prompt:

```
show apollo traffic
```


Following is sample output. Table 1-1 describes the fields.

```
Rcvd: 8 total, 0 format errors, 0 checksum errors, 0 bad hop count,
      8 local destination, 0 multicast
Bcast: 8 received, 0 sent
Sent: 16 generated, 0 forwarded
      0 encapsulation failed, 0 no route
      0 unknown
```

Table 1-1 Show Apollo Traffic Field Descriptions

Field	Description
format errors	Reported whenever a “bad packet” is detected (for example, corrupted header)
checksum errors	Should not be reported, as Apollo does not use a checksum
bad hop count	Increments when a packets hop count exceeds 16
encapsulation failed	Is registered when the router is unable to encapsulate a packet
unknown counter	Increments when packets are encountered that the router is unable to forward (for example, misconfigured helper-address, or no route available)

Displaying the Apollo ARP Table

Use the EXEC command **show apollo arp** to display that portion of the ARP table that pertains to the Apollo Domain Address Resolution Protocol. Enter this command at the EXEC prompt:

show apollo arp

Sample output follows:

```
Protocol  Address          Age (min)  Hardware Addr
Type      Interface
Apollo    123A.CAFE        -          0000.0c00.62e6
ARPA      Ethernet0
```

Debugging the Apollo Domain Network

Use the EXEC commands described in this section to troubleshoot and monitor the Apollo Domain network transactions. Generally, these commands are entered during troubleshooting sessions with Cisco engineers. For each **debug** command, there is a corresponding **undebug** command that turns the message logging off.

debug apollo-packet

The command **debug apollo-packet** outputs information about packets received, transmitted, and forwarded.

debug apollo-routing

The command **debug apollo-routing** prints out information on Apollo Domain routing packets.

Apollo Domain Global Configuration Command Summary

The following is an alphabetical list of the Apollo Domain global configuration commands, which specify system-wide parameters for Apollo Domain support.

[no] apollo access-list *name* {**permit**|**deny**} [*firstnet*]*lastnet.host* [*wildcard-mask*]

Specifies Apollo Domain access condition. The argument *name* is a name defined by the network administrator for the access list.

Choose the permit or deny condition for this list using the **permit** or **deny** keyword. The optional argument *firstnet* and the argument *lastnet.host* specify a selected network range. Use the argument *lastnet.host* to specify just one network. The optional *wildcard-mask* argument is a wildcard mask that uses the one bits to ignore the host part of the network address. Host bits corresponding to wildcard mask bits set to zero are used in comparisons. The **no** version of the command deletes the access list.

[no] apollo maximum-paths *paths*

Sets the maximum number of multiple paths that the router will remember and use. The argument *paths* is the number of paths to be assigned. The default value is one, which is restored with the **no** form of the command.

[no] apollo route *network network.address*

Specifies static routes for an Apollo Domain network. Packets received for the specified network will be forwarded to the specified router, whether or not that router is sending out dynamic routing. The **no** version of the command removes the routes.

[no] apollo routing *address*

Enables or disables Domain routing and specifies which system-wide host address to use. The argument *address* is a unique, five-digit hexadecimal host address.

Apollo Domain Interface Subcommand Summary

The following Apollo Domain interface subcommands specify line-specific parameters for Apollo Domain support. These subcommands must be preceded by an **interface** command.

[no] apollo access-group *name*

Specifies the interface on which an Apollo Domain access list is defined. Enter the user-defined *name* for the access list defined by the **apollo access-list** global configuration command for the argument *name*. The **no** version of the command removes the name.

[no] apollo network *number*

Assigns Apollo Domain network numbers to the appropriate interfaces. The argument *number* is an eight-digit hexadecimal number. The **no** version of the command removes a network number.

apollo update-time *seconds*

Sets the Apollo Domain routing update timers. The argument *seconds* specifies the interval between updates.

