Chapter 10
Routing AppleTalk

10

This chapter describes the routing process of the AppleTalk network protocol. The topics and tasks described in this chapter include:

- An overview of the AppleTalk routing protocol.
- Cisco's implementation of AppleTalk on both extended (also known as Phase II) and nonextended (Phase I) interfaces.
- Configuring AppleTalk routing.
- Configuring AppleTalk access list filters.
- Monitoring and debugging an AppleTalk network.

For more detailed information about the AppleTalk network systems, refer to the appendix "References and Recommended Reading."

Cisco's Implementation of AppleTalk

AppleTalk was designed as a client-server, or *distributed*, network system. In other words, users share network resources, such as files and printers, with other users. Interactions with servers are essentially transparent to the user, as the computer itself determines the location of the requested material, and accesses it without requesting information from the user.

AppleTalk identifies several network entities, of which the most elemental is a *node*. A node is simply any device connected to an AppleTalk network. The most common nodes are Macintosh computers and laser printers, but many other types of computers are also capable of AppleTalk communication, including IBM PCs, Digital VAX/VMS systems and a variety of workstations. A router is considered a node on each connected network. To avoid confusion, these router nodes are referred to as *ports*. Cisco routers support only one port per physical interface. The terms port and interface are used interchangeably in this document's discussion of AppleTalk routing. The next entity defined by AppleTalk is a *network*. An AppleTalk network is simply a single logical cable. Finally, an AppleTalk *zone* is a logical group of one or more (possibly noncontiguous) networks. These AppleTalk entities are shown in Figure 1-1.

Apple Computer has produced a variety of internetworking products with which to connect AppleTalk local-area networks. Apple supports Ethernet, Token Ring, FDDITalk, and its own proprietary twisted-pair media access system (called LocalTalk). However, to allow an AppleTalk network full participation in a multiprotocol internetwork, a multiprotocol router is required.

All routers from Cisco Systems support the AppleTalk network protocol (both extended and nonextended) over FDDI, Ethernet, Token Ring, synchronous serial, and X.25 interfaces.

Figure 1-1 AppleTalk Entities

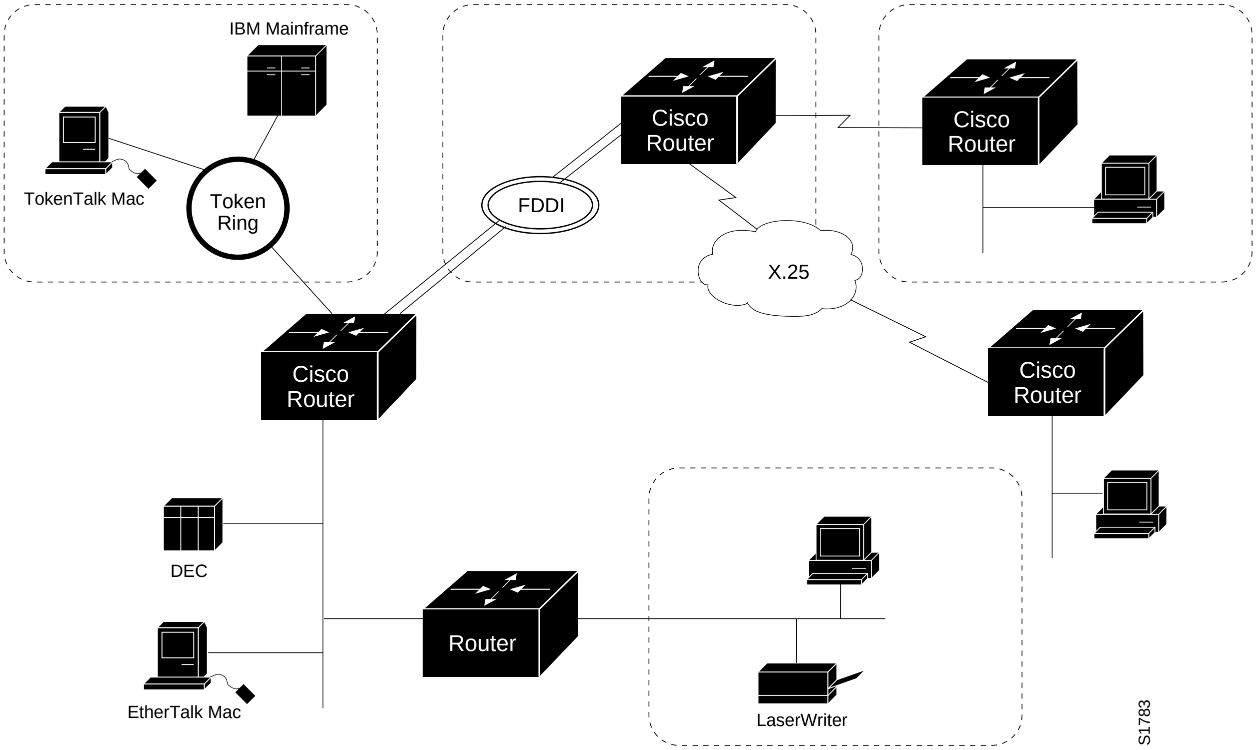


Figure 1-2 compares the AppleTalk protocols with the standard seven-layer OSI model, and illustrates how AppleTalk works with a variety of physical and link access mechanisms.

The Cisco AppleTalk implementation provides the following standard services, in addition to the ability to route any AppleTalk packet:

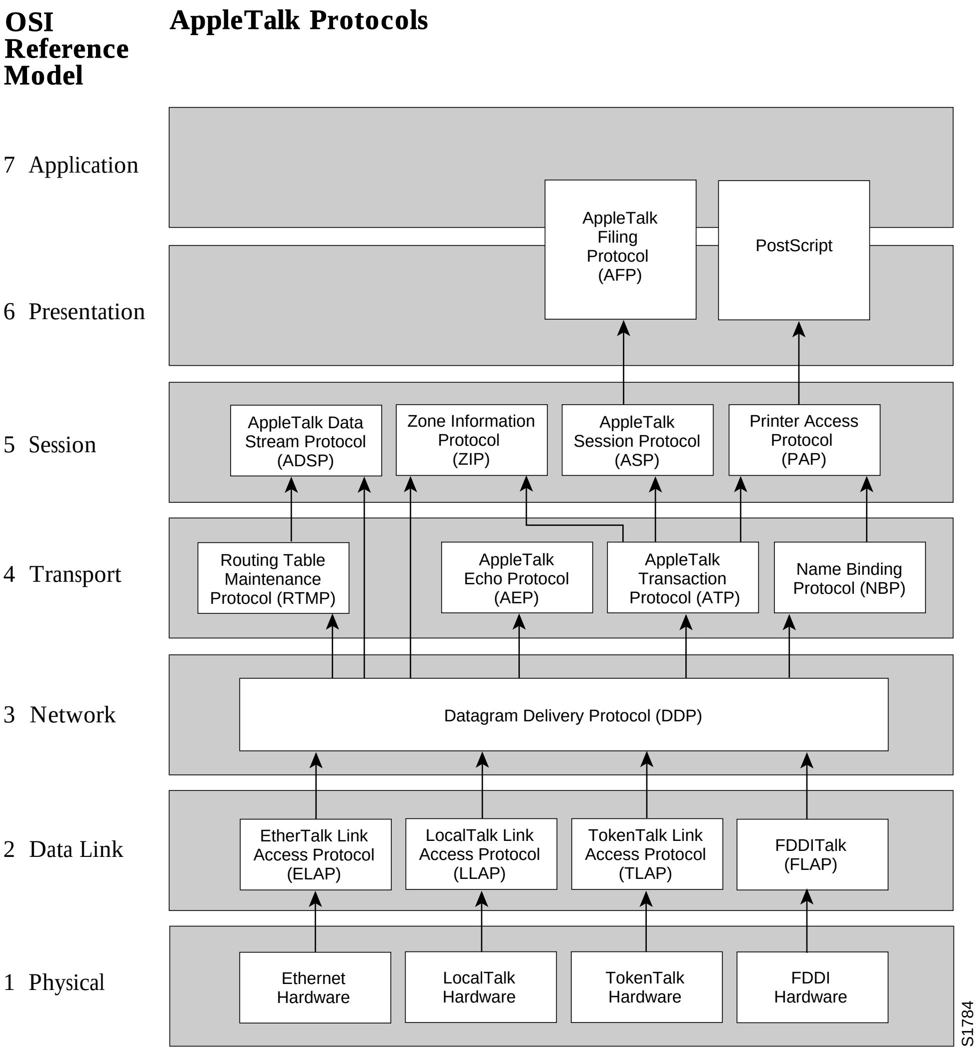
- AppleTalk Address Resolution Protocol (AARP)
- Datagram Delivery Protocol (DDP)
- Routing Table Maintenance Protocol (RTMP)
- Name Binding Protocol (NBP)
- AppleTalk Echo Protocol (AEP)
- AppleTalk Transaction Protocol (ATP)
- Zone Information Protocol (ZIP)

The Cisco AppleTalk implementation also includes the following enhancements:

- Support for EtherTalk 1.2 and EtherTalk 2.0 without the requirement for translation or transition routers
- Support for serial protocols, including SMDS, Frame Relay, X.25 and HDLC
- Configurable protocol constants
- No software limits on the number of zones or routes supported
- MacTCP support via the MacIP server
- NBP proxy service providing compatibility between the two AppleTalk standards
- IP encapsulation of AppleTalk, IPTalk, and Columbia AppleTalk Package (CAP) support
- Access control support to allow filtering of zones, routing data, and packets
- Integrated node name support to simplify AppleTalk management
- Interactive access to AEP and NBP provided via the ping router command
- Support for both configured (aka seed) and discovered port configuration
- Responder support used by *Inter•Poll*™ and other network monitoring packages
- SNMP over AppleTalk support

The DDP, RTMP, and AARP protocols provide end-to-end connectivity between internetworked nodes. NBP maps network names to AppleTalk internet addresses. NBP relies on ZIP to help determine which networks belong to which zones. File and print access is provided through AFP and PAP respectively, which work in concert with applications such as AppleShare and print servers.

Figure 1-2 AppleTalk and the OSI Reference Model


OSI Reference Model
AppleTalk Protocols
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
AppleTalk Filing Protocol (AFP)
PostScript
AppleTalk Data Stream Protocol (ADSP)
Zone Information Protocol (ZIP)
AppleTalk Session Protocol (ASP)
Printer Access Protocol (PAP)
Routing Table Maintenance Protocol (RTMP)
AppleTalk Echo Protocol (AEP)
AppleTalk Transaction Protocol (ATP)
Name Binding Protocol (NBP)
Datagram Delivery Protocol (DDP)
EtherTalk Link Access Protocol (ELAP)
LocalTalk Link Access Protocol (LLAP)
TokenTalk Link Access Protocol (TLAP)
FDDITalk (FLAP)
Ethernet Hardware
LocalTalk Hardware
TokenTalk Hardware
FDDI Hardware
S1784

Note: Apple Computer uses the name *AppleTalk* to refer the Apple Networking Architecture, whereas the actual transmission media used in AppleTalk Network are referred to as LocalTalk (Apple Computer's proprietary twisted-pair transmission medium for AppleTalk), TokenTalk (AppleTalk over Token Ring), EtherTalk (AppleTalk over Ethernet), and FDDITalk (AppleTalk over Fiber Distributed Data Interface).

AppleTalk, like many network protocols, makes no provisions for network security. The design of the AppleTalk protocol architecture requires that security measures be executed at higher application levels. Cisco Systems supports AppleTalk distribution lists, allowing control of routing updates on a per interface basis. It is a security feature similar to those provided for other protocols.

Extended (Phase II) Versus Nonextended (Phase I) AppleTalk

AppleTalk was designed for local work groups. With the installation of over 1.5 million Macintoshes in the first five years of the product's life, Apple found that some large corporations were exceeding the design limits of AppleTalk. Apple's solution was to create extended AppleTalk. The extended AppleTalk architecture increases the number of nodes per AppleTalk internetwork to over 16 million and an unlimited number of zones per cable. Apple also enhanced AppleTalk's routing capabilities and reduced the amount of network traffic generated by AppleTalk routers.

The introduction of the extended AppleTalk architecture also introduces the concept of *nonextended* and *extended* networks. Nonextended AppleTalk networks are sometimes called "Phase I," and extended networks are called "Phase II." Nonextended networks refer to the nonextended AppleTalk Ethernet 1.0 networks (explicitly removed by Apple but still supported by Cisco), and to the nonextended serial line-based networks, including those configured using X.25 and LocalTalk.

Extended networks refer to the extended AppleTalk-compliant networks configured on Ethernet (EtherTalk 2.0), FDDI, and Token Ring media. Samples of the AppleTalk nonextended and extended network configurations can be found in the section "AppleTalk Configuration Examples."

The AppleTalk extended-network architecture provides extensions compatible with nonextended AppleTalk internetworks. The AppleTalk extended architecture was designed to remove the previous limits of 254 concurrently active AppleTalk nodes per cable, as well as the previous limit of one AppleTalk zone name per cable. Extended AppleTalk contains better algorithms for choosing the best routers for traffic and is designed to minimize the amount of broadcast traffic generated for routing updates.

Another important feature in extended AppleTalk is the ability of a single AppleTalk cable to be assigned more than one network number. The size of the range of network numbers assigned to a cable determines the maximum number of concurrently active AppleTalk devices that can be supported on that cable, which is 254 devices per network number.

Cisco routers running software Release 8.2 or later support both extended and nonextended AppleTalk. Ethernet and serial interfaces may be configured for either extended or nonextended AppleTalk operation.

Note: Until every router in your internet supports AppleTalk Phase 2 (ATp2), you must observe the compatibility rules described in the "Configuration Guidelines (Compatibility Rules)" section of this chapter. Not all end-nodes must be upgraded to use the features provided by the AppleTalk enhancements.

Nonextended AppleTalk Addressing

AppleTalk addresses are 24 bits long. They consist of two components: a 16-bit network number, and an 8-bit node number. The Cisco AppleTalk software parses and displays these addresses as a sequence of two decimal numbers, first the network number, then the node number, separated by a dot. For example, node 45 on network 3 is written as 3.45. A node is any AppleTalk-speaking device attached to the network. Each enabled AppleTalk interface on a router is a node on its connected network.

AppleTalk Zones

When a router is used to join two or more AppleTalk networks into an internetwork, the component physical networks remain independent of each other. A network manager may assign to these network conceptual groupings known as *zones*.

There are two main reasons to create zones in an AppleTalk internetwork: to simplify the process of locating and selecting network devices, and to allow for the creation of departmental work groups that may exist on several different and possibly geographically separated networks.

For example, consider a large AppleTalk internetwork that may contain hundreds or thousands of shared resources and devices. Without a method of dividing this large number of resources and devices into smaller groups of devices, a user might have to scroll through hundreds or thousands of resource/device names in the Chooser to select the one resource to be used. By creating small, conceptual groups of resource and device names, a user may now choose the resource they need much more quickly and easily than if they were sorting through a very long list of names.

A zone may include many networks, which need not be physically co-located. A zone is not limited by geographical area.The partitioning afforded by zone names is conceptual, not physical.

Zones are defined by the network manager during router configuration. When a Cisco router is configured, each AppleTalk-configured interface must be associated with exactly one zone name for nonextended networks, or one or more zone names for extended networks. Until a zone name has been assigned, AppleTalk routing features are disabled for that interface. The section “Configuring AppleTalk Routing” later in this chapter describes the subcommands to use in the zone-naming process.

It is very important that routers explicitly configured with zone information be configured correctly.

Name Binding Protocol (NBP)

The Name Binding Protocol (NBP) maps network entity names to internetwork addresses. It allows users to specify descriptive or symbolic names, while other software processes refer to numerical addresses for the same entities. With NBP, almost all user-level programs respond to names instead of numbers. When users select an AppleTalk device, they are using the NBP protocol to translate the device's entity name to the entity's network address. Numerical addresses dynamically assigned to nodes are primarily used by the router software and by network managers in the ping process (see the section "The AppleTalk Ping Command" later in this chapter).

NBP provides four basic services for binding names to nodes and zones:

- Name registration
- Name deletion
- Name look-up
- Name confirmation

The nature of the AppleTalk addressing scheme is inherently volatile and node addresses change frequently. Therefore, NBP associates numerical addresses with aliases that continue to reference the correct address if the address changes.

Zone Information Protocol (ZIP)

NBP uses the Zone Information Protocol (ZIP) to determine which networks belong to which zones. A Cisco router uses ZIP to maintain the network-number-to-zone-name mapping of the AppleTalk internet.

Along with a routing table, each router maintains a data structure known as the *zone information table* (ZIT). The table provides a listing of network numbers for each network in every zone. Each entry is a *tuple* (an inseparable network-number-hop-number set) that matches a network number with a zone name as supplied by the network manager.

Dynamic Configuration (Discovery Mode)

AppleTalk provides for *dynamic configuration*. With dynamic configuration, not all fields of an AppleTalk address need to be specified in the configuration of a router. If there is another AppleTalk router on the network, it may be able to supply the network number and zone name. A preconfigured router on an AppleTalk network acts as a *seed router,* responding to configuration queries from other routers on its connected network.

Seed routers are routers that come up and verify the configuration with an operational router. If the configuration is valid, they start functioning. Seed routers come up even if no other routers are on the network. On the other hand, a *nonseed router* must first communicate with a seed router before it can commence operation. A nonseed router must obtain and verify the configuration with another functioning router. The configuration of the nonseed router must match exactly with the configuration of the seed router for the nonseed router to function.

An end node always behaves in a manner similar to discovery mode. It uses any previous configuration as a starting point for initialization.

Unspecified parts of the AppleTalk address are entered as zero. Table 1-1 illustrates AppleTalk addresses that feature unspecified addressing.

Table 1-1 Examples of AppleTalk Addresses

AppleTalk Address	Description
34.5	Represents a fully qualified address (net 34, node 5)
0.5	A partially qualified address (net unspecified, node 5)
122.0	Represents net 122, node unspecified
0.0	Address is completely unspecified

Node numbers are automatically assigned by AppleTalk configured as zero. When the specified address is in use the node randomly chooses its node number. The node will first try the node number that was its most recent address. If that number is unavailable, the node then searches for the next available address. If it reaches 254 without finding an available number, it cycles back to 1 and continues until it finds a free address. LocalTalk address restrictions are as follows: user node numbers are from 1 to 127, and server/printer node numbers are from 128 to 254. Nonextended Ethernet and extended media do not observe the server/user node distinction. The protocol reserves node numbers 0 and 255. Extended media also reserves the node number of 254.

For nonseed routers, an interface will behave as an AppleTalk end node. If zero has been specified for a network number, that interface will not route any packets until it receives its network number from a seed router.

Receipt of a routing table update informs the router of the network number for the interface on which the packet was received. Every routing table update includes the network number of the network the packet was sent on. Therefore, the router is able to determine the network number of the receiving interface.

As long as one fully configured router exists on a physical network segment (or *cable*), other routers directly attached to that cable may use discovery mode to determine their configuration; they can take their information from an operational router. However, once the configuration process has stabilized for a particular AppleTalk internet, all routers thereafter should be configured as seed routers. Note that synchronous X.25 network interfaces must be explicitly configured on each router to be used as AppleTalk transports.

RTMP routing tables contain an entry for every network in the internet. Each entry includes the router port which leads to the destination network, the node ID of the next router to receive the packet, and the distance in hops to the destination network. Periodic exchange of routing tables allows the routers in an internet to ensure accurate and consistent information.

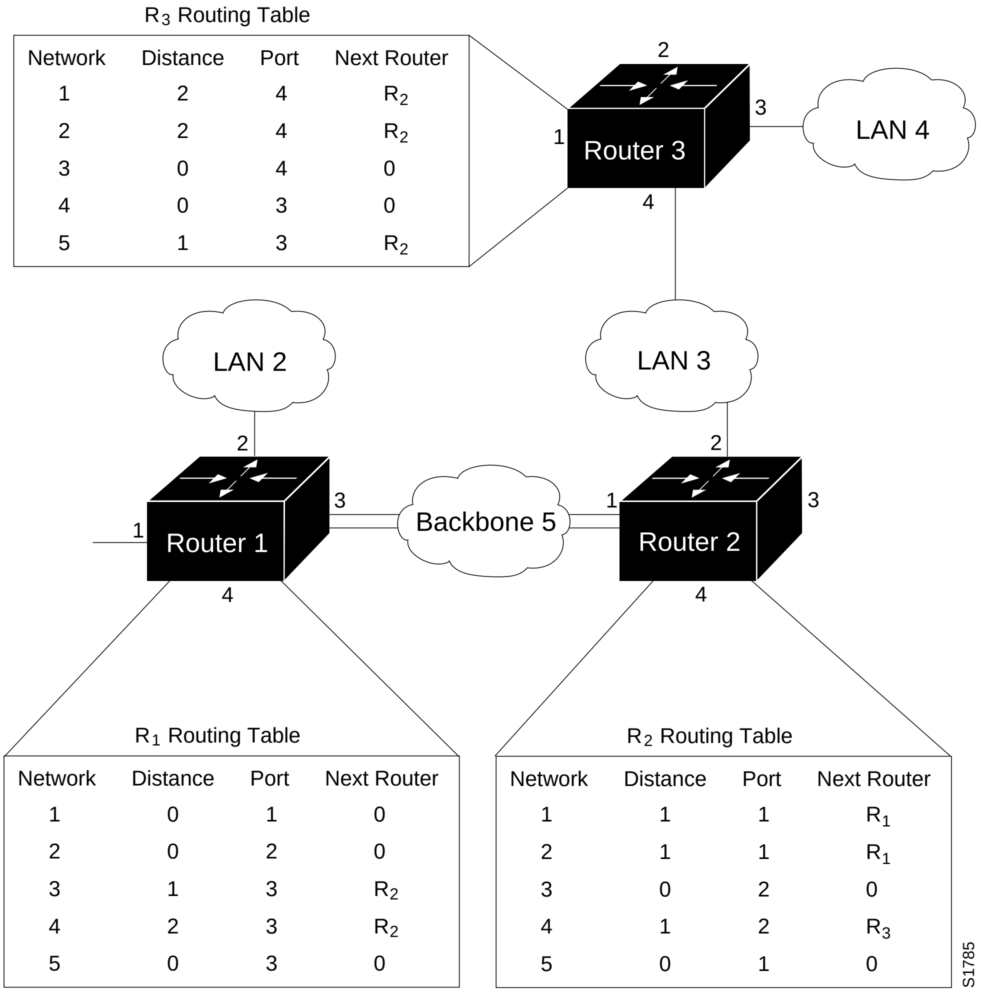
Node address information is maintained by tables appropriate to the media (usually AARP tables).

Figure 1-3 shows a sample RTMP table and the corresponding network topology.

Extended AppleTalk Addressing

AppleTalk addresses, as explained in the section “Nonextended AppleTalk Addressing,” earlier in this chapter, are composed of a 16-bit network and an 8-bit node number. In nonextended AppleTalk, nodes within a single cable can communicate using only their 8-bit node numbers.

Figure 1-3 Sample AppleTalk Routing Table



R_3 Routing Table

Network	Distance	Port	Next Router
1	2	4	R_2
2	2	4	R_2
3	0	4	0
4	0	3	0
5	1	3	R_2

R_1 Routing Table

Network	Distance	Port	Next Router
1	0	1	0
2	0	2	0
3	1	3	R_2
4	2	3	R_2
5	0	3	0

R_2 Routing Table

Network	Distance	Port	Next Router
1	1	1	R_1
2	1	1	R_1
3	0	2	0
4	1	2	R_3
5	0	1	0

A node in extended AppleTalk is always identified by its network and node number. Dynamic address resolution when a router is not present includes the assignment of a random network number within a small range, as well as a node number. When a router is present in the network, a node starts up using its newly acquired address for a short period of time. It then immediately requests the range of valid network numbers from an operational router. The node then uses this to determine its actual AppleTalk address by selecting an unassigned address.

A new concept of cable *ranges* is introduced with the extended AppleTalk. Cables now have ranges of network numbers and multiple zones that may exist on them, so that a node can access anything that is in any of the zones that are on the same cable as the node itself. But the node can exist in only one zone and on only one network.

In an extended AppleTalk network, the mapping of a physical cable to a zone name is no longer valid. End nodes are expected to know the zone to which they belong, or to choose from the list of available zones provided by a router. The router maintains a default zone that new nodes will use automatically if they have not previously chosen a zone.

AppleTalk Name Registration

Cisco routers with active AppleTalk interfaces register each interface separately. A unique interface name is generated by appending the interface type name and unit number to the router name. For example, if a router is named myrouter, and has Appletalk enabled on Ethernet 0 in zone Engineering, the NBP registered name will be:

```
myrouter.Ethernet0:ciscoRouter@Engineering
```

The NBP name is deregistered in the event that AppleTalk is disabled on an interface by configuration or due to interface errors.

Registering each interface on the router provides the AppleTalk site administrator with a positive indication that each interface on the router is properly configured and operating.

One name is registered per interface; other service types are registered once for every zone name on the router. The following display output from a **show appletalk nbp** command illustrates this. This display shows that each interface is uniquely identified, but that only one SNMP Agent is generated per zone.

```
  Net Adr Skt Name                    Type            Zone
 4042   8 254 sloth.Ethernet3         ciscoRouter     Engineering
 4042   8   8 sloth                   SNMP Agent      Engineering
 4028   8 254 sloth.Ethernet2         ciscoRouter     Engineering
 4035   8 254 sloth.TokenRing0        ciscoRouter     Engineering
 4036   8 254 sloth.TokenRing1        ciscoRouter     Engineering
 4038   8 254 sloth.Hssi0             ciscoRouter     Narrow Beam
 4038   8   8 sloth                   SNMP Agent      Narrow Beam
```

AppleTalk Responder Support

The router answers Appletalk *responder* requests. The *listener* is installed on the Appletalk interface name registration socket.

The response packet generated supplies the bootstrap firmware version string, followed by the router operating software version string. These are displayed in the position of the Macintosh System version and the Macintosh printer driver version, respectively, in such applications as Apple's *Inter•Poll*™.

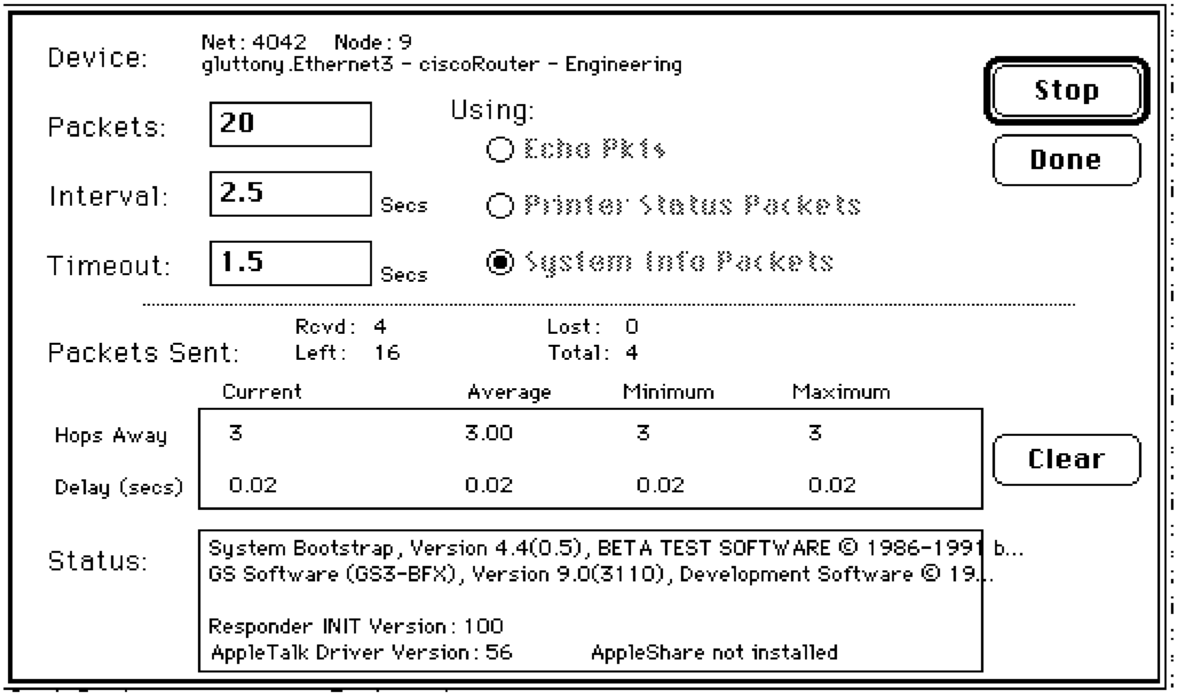
The response packet contains strings similar to those displayed by the **show version** EXEC command.

The information returned is as follows:

- The system bootstrap version (ROM version).
- The currently running software version.
- The AppleTalk version—This always indicates 56, which is the first Apple Macintosh version that contained AppleTalk Phase 2 support.
- The AppleTalk responder version—This always displays 100, which indicates support of Version 1.0 responder packets.
- Finally, the Cisco router reports that AppleShare is not installed.

Figure 1-4 illustrates a typical output display for *Inter•Poll* that lists this information.

Figure 1-4 Sample illustration of *Inter•Poll* Output



Device: Net: 4042 Node: 9
gluttony.Ethernet3 - ciscoRouter - Engineering

Packets: 20

Interval: 2.5 Secs

Timeout: 1.5 Secs

Using:
○ Echo Pkts
○ Printer Status Packets
● System Info Packets

Packets Sent: Rcvd: 4 Left: 16 Lost: 0 Total: 4

	Current	Average	Minimum	Maximum
Hops Away	3	3.00	3	3
Delay (secs)	0.02	0.02	0.02	0.02

Status:

```
System Bootstrap, Version 4.4(0.5), BETA TEST SOFTWARE © 1986-1991 b...
GS Software (GS3-BFX), Version 9.0(3110), Development Software © 19...

Responder INIT Version: 100
AppleTalk Driver Version: 56        AppleShare not installed
```

Configuring AppleTalk Routing

This section provides an overview on how to configure Cisco AppleTalk routing.

Configuration Overview

The AppleTalk interface configuration is different for the two types of AppleTalk interfaces: extended and nonextended.

Configuring a nonextended AppleTalk interface involves the following steps:

Step 1: Enable AppleTalk routing with the **appletalk routing** command.

Step 2: Assign the nonextended AppleTalk addresses with the **appletalk address** interface subcommand.

Step 3: Assign the zone name with the **appletalk zone** interface subcommand.

Configuring an extended AppleTalk interface involves these steps:

Step 1: Enable AppleTalk routing with the **appletalk routing** command.

Step 2: Assign the extended AppleTalk cable range parameters with the **appletalk cable-range** command.

Step 3: Assign the zone name or names with the **appletalk zone** interface subcommand.

The software also provides commands for fine tuning the AppleTalk network, for configuring packet filtering mechanisms, monitoring, maintaining and troubleshooting network operation. Alphabetically arranged summaries of the commands described in this chapter are also provided at the end of the chapter.

Configuration Guidelines (Compatibility Rules)

Follow these guidelines when configuring your AppleTalk network on a Cisco router:

- If your AppleTalk internet contains any routers that support only nonextended AppleTalk, the following configuration restrictions must be observed. These restrictions are not enforced, but unpredictable behavior may result if they are violated. All routers in a network must support extended AppleTalk before these restrictions may be lifted.
  - Cable ranges of only one (*666-666*, for example) are permitted.
  - Each AppleTalk network may have only one zone associated with it.

Follow these guidelines when using Cisco routers with other vendors' AppleTalk implementations:

- A Macintosh that contains an Ethernet card must run EtherTalk Version 2.0 or later to support extended AppleTalk. A Macintosh with only a LocalTalk interface does not require any changes.

- Shiva FastPath routers must run K-Star Version 8.0 or later and be explicitly configured for extended AppleTalk.
- Apple's Internet Router software, Version 2.0, supports a transition mode for translation between the nonextended AppleTalk and the extended AppleTalk on the same network. Transition mode requires the Apple upgrade utility and a special patch file from Apple.

A general understanding of Cisco's representation of AppleTalk addresses is necessary before configuration of the router. Refer to the sections "Cisco's Implementation of AppleTalk," "Nonextended AppleTalk Addressing," and "Extended AppleTalk Addressing" earlier in this chapter.

Enabling AppleTalk Routing

Before you can configure AppleTalk routing, enable AppleTalk protocol processing. To do that, use the **appletalk routing** global configuration command. The full command syntax follows:

appletalk routing
no appletalk routing

The **appletalk routing** configuration command enables AppleTalk protocol processing. The **no appletalk routing** disables all AppleTalk processing.

Assigning Nonextended (Phase I) AppleTalk Address

To assign AppleTalk addresses for nonextended networks, use the **appletalk address** interface subcommand. Its full syntax follows.

appletalk address *address*
no appletalk address

The argument *address* assigns AppleTalk addresses on the interfaces that will be used for the AppleTalk protocol. It assigns one AppleTalk address per interface. This step must be done before assigning zone names.

Note: Use this subcommand to configure nonextended interfaces.

The **no appletalk address** subcommand disables nonextended AppleTalk processing on the interface.

Example:

These commands begin AppleTalk routing and assign address 1.129 to interface Ethernet 0.

```
appletalk routing
!
interface ethernet 0
appletalk address 1.129
```

Assigning a Cable Range for Extended AppleTalk (Phase II)

To assign the cable-range parameters, use the **appletalk cable-range** interface subcommand. The full command syntax follows.

appletalk cable-range *start-end* [*network.node*]
no appletalk cable-range *start-end* [*network.node*]

This command designates an interface to be on an extended AppleTalk network. A cable range is the network numbers assigned to an extended network.

This range is specified using the argument *start-end*, which is a pair of decimal numbers between 1 and 65,279, inclusive. The starting network number should be less than or equal to the ending network number.

Specifying a cable range of 0-0 in the *start-end* argument (start = end = 0) places the interface into discovery mode, which attempts to determine cable range information from another router on that network.

The optional *network.node* argument specifies the suggested network and node number that will be used first when selecting the AppleTalk address for this interface. Note that any suggested network number must fall within the specified range of network numbers.

Use the **no appletalk cable-range** command to disable AppleTalk processing on the interface.

Example:

This command assigns a cable range of 2-2 to the interface:

```
appletalk cable-range 2-2
```

Assigning a Zone Name

Use the **appletalk zone** interface subcommand to assign a zone name to an AppleTalk interface. Full command syntax for this command follows:

appletalk zone *zonename*
no appletalk zone [*zonename*]

Interfaces that are configured for seed routing or that have discovery mode disabled must have a zone name assigned before AppleTalk processing will begin.

The argument *zonename* specifies the name of the zone for the connected AppleTalk network. The argument *zonename* may include special characters from the Apple Macintosh character set. To include a special character, insert a colon and two uppercase hexadecimal characters. The hexadecimal equivalent for special characters in the Macintosh character set may be found in character tables published by Apple Computer (see Appendix D in the text *Inside AppleTalk*, 2nd edition).

Note: Due to restrictions associated with Cisco's configuration parsing, it is not possible to define zone names with leading or trailing space characters. Although permitted by the standard, such names are not recommended due to the potential confusion that can be caused for users.

The **appletalk zone** command is used with both extended and nonextended configurations. Extended configurations may repeat this command to define a list of zones for the network.

The first zone specified in the list is the *default zone.* The router always uses the default zone when registering NBP names for interfaces. Computers in the network will select the zone in which they will operate from the list of zone names valid on the cable to which they are connected. If an interface is using nonextended AppleTalk, repeated execution of the zone command will replace the zone name for the interface with the newly specified zone name.

The **no appletalk zone** interface subcommand deletes a zone name from a zone list or the entire zone list if none is specified. The optional zone name is ignored for nonextended AppleTalk interface configurations. The command is also ignored if the specified zone name is not in the current zone list for an interface. The list should be cleared using the **no appletalk zone** interface subcommand before configuring a new zone list.

Note: The zone list is cleared automatically when **appletalk address** or **appletalk cable-range** commands are used. Additionally, the zone list is cleared if the **appletalk zone** command is used on an *existing* network; this can occur when adding zones to a set of routers until all routers are in agreement.

Examples:

This command assigns the zone name Twilight to an interface:

```
appletalk zone Twilight
```

The following example shows use of the AppleTalk special characters sets by setting the zone name to *cisco•zone.*

```
appletalk zone cisco:A5zone
```

Setting and Resetting Discovery Mode

Discovery mode is set using the **appletalk discovery** interface subcommand. The full syntax of this command follows:

appletalk discovery
no appletalk discovery

This command resets the discovery mode and allows a new cable range to be discovered. If the port information has been discovered, and the port is operational, then this command results in the port being a valid seed port.

Use the **no appletalk discovery** command to return the software to the default (off) state.

Use the command **no appletalk discovery** to allow the interface to be a seed port. If the interface is not operational when this command is issued, you must configure the zone names before the interface will be operational. Otherwise, the current zone list is retained as part of the configuration.

Using Discovery Mode

Cisco's implementation of discovery mode is compliant with the mechanism defined by Apple. The network definition for a router using discovery mode is confirmed, or modified to match the network configuration known by a seed router. The router in discovery mode then learns the associated zones from that router and the port becomes operational. A seed router is required on each network.

While the port is operational, it acts like a seed router for any other routers that come on-line. However, another operational router port is still needed if the first port is restarted for any reason.

Note: It is not advisable to have all routers on a network configured with discovery mode enabled. If all routers restart simultaneously (for instance, after a power failure), the network is inaccessible until discovery mode is manually stopped via operator intervention.

Discovery mode is particularly useful while changing a network configuration or when adding a router to an existing network.

Configuring IP Encapsulation of AppleTalk Packets

Use the **appletalk iptalk** interface subcommand to encapsulate AppleTalk in IP packets in a manner compatible with the CAP IPTalk and the Kinetics IPTalk (KIP) implementations.

appletalk iptalk *net.node zone*
no appletalk iptalk

This command enables IPTalk encapsulation on an interface which already has an configured IP address. The command allows AppleTalk communication with UNIX™ hosts running older versions of CAP which do not support native AppleTalk EtherTalk encapsulations. Typically, Apple Macintosh users wishing to communicate with these servers would have their connections routed through a Kinetics FastPath™ router running KIP (Kinetics IP) software.

Use the **no appletalk iptalk** command to disable IPTalk encapsulation on the interface.

This command is provided as a migration command; newer versions of CAP provide native AppleTalk EtherTalk encapsulations and the IPTalk encapsulation is no longer required. The Cisco implementation of IPTalk assumes that AppleTalk is already being routed on the backbone, since there is currently no LocalTalk hardware interface for Cisco routers.

The Cisco implementation of IPTalk does not support manually configured AppleTalk-to-IP address mapping (atab). The address mapping provided is the same as the Kinetics IPTalk implementation when the atab facility is not enabled. This address mapping functions as follows: The IP subnet mask used on the router Ethernet interface on which IPTalk is enabled is inverted (one's complement). This result is then masked against 255 (0xFF hexadecimal). This is then masked against the low-order 8 bits of the IP address to obtain the AppleTalk node number. The following example configuration should make this more clear.

Example:

```
interface Ethernet 0
ip address 131.108.1.118 255.255.255.0
appletalk address 20.129
appletalk zone Native AppleTalk
appletalk iptalk 30.0 UDPZone
```

In this configuration, the IP subnet mask would be inverted:

```
255.255.255.0 inverted yields: 0.0.0.255
```

Masked with 255 it yields 255, and masked with the low-order 8 bits of the interface IP address it yields 118.

This means that the AppleTalk address of the Ethernet 0 interface seen in the UDPZone zone is 30.118. This caveat should be noted, however: Should the host field of an IP subnet mask for an interface be more than 8 bits wide, it will be possible to obtain conflicting AppleTalk node numbers. For instance, consider a situation where the subnet mask for the Ethernet 0 interface above is 255.255.240.0, meaning that the host field is 12 bits wide.

Configuring IP Encapsulation DDP Socket to UDP Port Mapping

Use the global configuration subcommand **appletalk iptalk-baseport** to specify the UDP port number that is the beginning of the range of UDP ports used in mapping AppleTalk *well-known* DDP socket numbers to UDP ports. The command syntax looks like this:

appletalk iptalk-baseport *port-number*
no appletalk iptalk-baseport

Use the **no** form of the **appletalk iptalk-baseport 768** command to return to the default setting.

Implementations of IPTalk prior to April 1988 mapped well-known DDP socket numbers to privileged UDP ports starting at port number 768. In April of 1988, the NIC assigned a range of UDP ports for the defined DDP well-known sockets starting at UDP port number 200 and assigned these ports the names *at-nbp*, *at-rtmp*, *at-echo* and *at-zis*. The CAP, Release 6 and later dynamically decides which port mapping to use. If there are no AppleTalk service entries in the */etc/services* file, CAP will use the older 768-based mapping.

This is the default UDP port mapping supported by Cisco's implementation of IPTalk. If there are service entries in the */etc/services* file for the AppleTalk services, the Cisco router configured for IPTalk encapsulation should specify the beginning of the port mapping range with the **appletalk iptalk-baseport** command. The following example configuration builds upon the example for the **appletalk iptalk** command to illustrate this concept.

Example:

```
appletalk iptalk-baseport 200
!
interface Ethernet 0
ip address 131.108.1.118 255.255.255.0
appletalk address 20.129
appletalk zone Native AppleTalk
appletalk iptalk 30.0 UDPZone
```

Checking Packet Routing Validity

Use the **appletalk strict-rtmp** global configuration command to enforce maximum checking of routing packets to ensure their validity. The full command syntax follows:

appletalk strict-rtmp
no appletalk strict-rtmp

The default of this command is to provide maximum checking.

Currently, strict RTMP checking consists of discarding RTMP packets arriving from routers that are not directly connected to the router performing the check. In other words, no routed RTMP packets will be accepted.

Use the **no appletalk strict-rtmp** command to disable the maximum-checking mode.

Enabling and Disabling Routing Updates

Use the interface subcommand **appletalk send-rtmp** to allow the transmission of routing updates to be disabled for a specific interface. The full syntax of the command is:

appletalk send-rtmp
no appletalk send-rtmp

This command allows a router to be placed on a network with AppleTalk routing enabled, without being seen by other AppleTalk routers on the cable. The default is to send routing updates. The **no appletalk send-rtmp** command disables this default.

Changing Routing Timers

Use the global configuration command **appletalk timers** to change the time intervals used in AppleTalk routing, as follows:

appletalk timers *update-interval valid-interval invalid-interval*
no appletalk timers *update-interval valid-interval invalid-interval*

The argument *update-interval* is the time, in seconds, between routing updates sent to other routers on the network. This is ten seconds by default. A route is considered suspect anytime it is older than the specified *update-interval*.

The argument *valid-interval* is amount of time, in seconds, that the router will consider a route valid without having heard a routing update for that route. This is normally twice the update interval, 20 seconds by default. Once this period of time has elapsed without having heard a routing update for a route, the route becomes bad, and is now eligible for replacement by a path with a higher (less favorable) metric.

The argument *invalid-interval* is the amount of time, in seconds, that the route is retained after being marked as bad. During this period, routing updates include this route with a special *notify neighbor* metric. If this timer expires, the route is deleted from the routing table. By default, this timer is three times the valid interval, or 60 seconds.

Any of the timers may be specified as zero to specify the system default value.

Note: Modification of the routing timers should not be undertaken without fully understanding the ramifications of doing so. Many other AppleTalk router vendors provide no facility for modifying their routing timers; should you adjust a Cisco router's AppleTalk timers such that routing updates do not arrive at these other routers within the normal interval, it is possible to degrade or destroy AppleTalk network connectivity.

Example:

This command increases the update interval to 20 seconds, the route-valid interval to 40 seconds, and the route-invalid interval to 60 seconds.

```
appletalk timers 20 40 60
```

Assigning a Proxy Network Number

When an AppleTalk internetwork contains routers that support only nonextended AppleTalk and routers that support only extended AppleTalk, then one **appletalk proxy-nbp** global configuration command is required for each zone in which there is a router that supports only nonextended AppleTalk. The full syntax of this command follows.

appletalk proxy-nbp *network-number zonename*
no appletalk proxy-nbp *network-number zonename*

The argument *network-number* must be a unique network number that will be advertised via this router as if it were a real network.

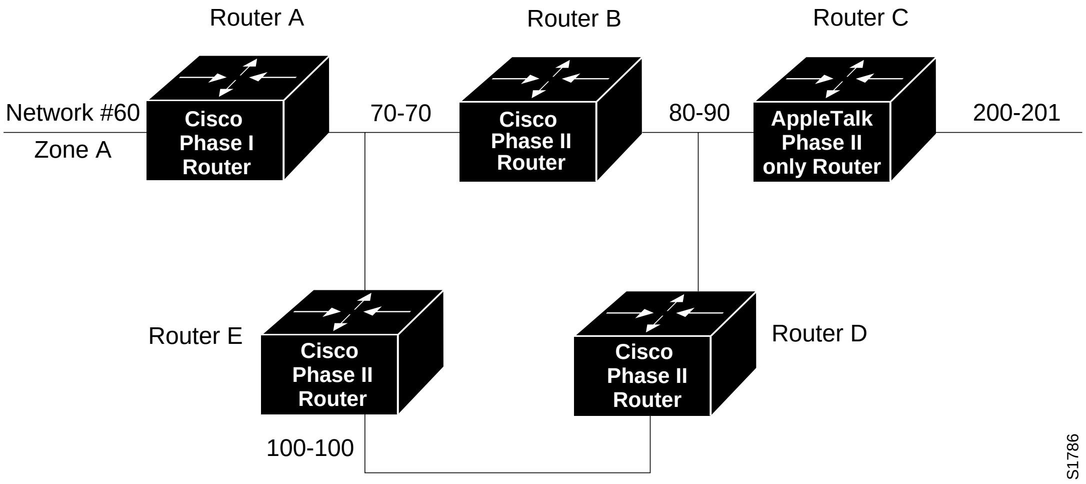
The argument *zonename* is the name of the zone requiring compatibility support.

No router may have the same network number defined as a proxy network, and no network number can be associated with a physical network.

Only one proxy is needed to support a zone, but additional proxies may be defined with different network numbers if redundancy is desired. Each proxy will generate one or more packets for each forward request it receives. All other packets sent to the proxy network are discarded. Redundant proxies increase the NBP traffic linearly.

Assume your network topology looks like the one in Figure 1-5. Also assume that Router A supports only nonextended AppleTalk, that Router B supports only extended AppleTalk (not in transition mode), and that Router C supports only extended AppleTalk.

Figure 1-5 Example Network Topology



If Router C generates a NBP hookup request for zone A, router B will convert this request to a forward request and send it to Router A. Since Router A supports only nonextended AppleTalk, it does not handle the forward request and ignores it. Hence, the NBP lookup from Router C fails.

To work around this problem without putting a transition router adjacent to the nonextended only router (Router A), you could configure Router D with a NBP proxy.

If you configured router D with a NBP proxy as follows, any forward requests received for Zone A are converted into lookup requests, and therefore, the nonextended router for Net 60 can properly respond to NBP hookup requests generated beyond Router C. The following example demonstrates the command needed to describe this configuration.

Example:

```
appletalk proxy 60 A
```

Generating Checksum Verification

Use the **appletalk checksum** global configuration command to enable the generation and verification of checksums for all AppleTalk packets. The command syntax follows:

appletalk checksum
no appletalk checksum

An incoming packet with a nonzero checksum will be verified against that checksum and discarded if in error. By default, checksum verification is enabled.

Cisco routers no longer check checksum on routed packets, thereby eliminating the need to disable checksum to allow operation of some networking applications.

Use the **no appletalk checksum** command to disable checksum verifications.

Specifying the Time Interval Between AppleTalk ARP Transmissions

Use the **appletalk arp interval** global configuration command to specify the time interval between retransmission of ARP packets, as follows:

appletalk arp {**request**|**probe**} **interval** *milliseconds*
no appletalk arp

The argument *milliseconds* specifies the interval. The minimum value is 33 milliseconds. The defaults for the *milliseconds* argument depend on the **probe** and **request** keywords as follows:

- **probe**—*milliseconds* = 200
- **request**—*milliseconds* = 1000

The keywords **request** and **probe** have the following effects:

- The **interval** *millisecond* value specified with the **request** version of this command is used when AARP is attempting to determine the hardware address for a different node, so a packet can be delivered. These **interval** *millisecond* values may be changed as desired although the defaults are optimal for most sites.
- The **interval** *millisecond* value specified with the **probe** version is used when obtaining the address of this router (router being configured). These **interval** *millisecond* values should not be changed from the defaults, since they directly modify the dynamic node assignment algorithm.

Lengthening the interval between packets permits responses from certain devices that respond more slowly, such as printers and overloaded file servers, to be received.

All values take effect immediately and are global to the router. The current values are available using the **show appletalk global** EXEC command.

The command **no appletalk arp** or a *milliseconds* value of 0 resets the defaults.

Example:

This command lengthens the AppleTalk ARP retry interval to 2000 milliseconds.

```
appletalk arp request interval 2000
```

Specifying the AARP Retransmission Count

Use the **appletalk arp retransmit-count** global configuration command to specify the number of retransmissions that will be done before abandoning address negotiations and using the selected address.

appletalk arp {**request**|**probe**} **retransmit-count** *count*
no appletalk arp

The argument *count* specifies the retransmission count. The minimum value that can be specified is 1. The defaults for the *count* argument depend on the **probe** and **request** keywords as follows:

- **probe**—*count* = 10
- **request**—*count* = 5

The keywords **request** and **probe** have the following effects:

- The **retransmit-count** *count* value specified with the **request** version of this command is used when AARP is attempting to determine the hardware address for a different node, so a packet can be delivered. These **retransmit-count** *count* values may be changed as desired although the defaults are optimal for most sites.

- The **retransmit-count** *count* value specified with the **probe** version is used when obtaining the address of this router (router being configured). These **retransmit-count** *count* values should not be changed from the defaults, since they directly modify the dynamic node assignment algorithm.

All values take effect immediately and are global to the router. The current values are available using the **show appletalk global** EXEC command.

The command **no appletalk arp** or a *count* value of 0 resets the defaults.

Example:

This command specifies an AARP retransmit count of 10.

```
appletalk arp request retransmit-count 10
```

AppleTalk MacIP Routing and IP Address Management Service

Cisco routers allow for the routing of IP datagrams to IP clients using DDP as a low-level encapsulation—commonly referred to as *MacIP*.

The MacIP address management and routing services available in Cisco routers are described in detail in Draft Internet RFC, *A Standard for the Transmission of Internet Packets Over AppleTalk Networks*.

Some situations may *require* the use of MacIP. For example, when the attachment media used by Macintoshes does not have device driver support for IP. Cisco and Apple currently support attachment media that do not have native IP software drivers for Macintoshes, but where AppleTalk drivers are available. If your network falls into this category, then configuring MacIP services may provide a simple solution to IP connectivity requirements for Macintoshes.

Another case where MacIP services may be advantageous is in managing IP address allocations for a large, dynamic Macintosh population. There are several advantages to using the MacIP approach in this situation:

- Macintosh TCP/IP drivers can be configured in a completely standard way, regardless of location. Essentially, the dynamic properties of Appletalk address management become available for IP address allocation.
- All global parameters, such as IP subnet mask, Domain Name Services, and default routers, can be modified by the administrator in the Cisco Router. Macintosh IP users receive the updates by merely restarting their local TCP/IP driver.
- The administrator can monitor MacIP address allocations and packet statistics remotely by using the Telnet application to attach to the Cisco router console. This allows central administration of IP allocations in remote locations. For Internet sites, it allows remote technical assistance.

However, when evaluating the implementation of MacIP on a Cisco router, there are several considerations to weigh:

- Each packet from a Macintosh client destined for an IP host, or from an IP host destined for the Macintosh client, *must* pass through the Cisco router, if the client is using the router as a MacIP server. This increases traffic through the router in cases where the Cisco router is not a necessary hop. There is also a slight increase in router CPU use that is proportional to the number of packets delivered to and from active MacIP clients.
- Memory usage increases in the Cisco router, proportional to the total number of active MacIP clients (about 80 bytes per client).
- If you are using MacIP to allow Macintoshes to communicate with IP hosts on the same LAN segment (that is, the Macintoshes are on the Cisco interface on which MacIP is configured) and the IP hosts have extended IP access lists, these access lists should include entries to permit IP traffic destined for these IP hosts (from the MacIP addresses). If these entries are not present, packets destined for IP hosts on the local segment will be blocked (that is, they will not be forwarded).

Exceptions to Draft RFC

The Cisco implementation of MacIP confirms to the September 1991 draft RFC for MacIP, with the following exceptions:

- Fragmentation of IP datagrams that exceed the DDP MTU and that are bound for DDP clients of MacIP, is not performed.
- Routing to DDP clients outside of configured MacIP client ranges is not performed.

Configuring MacIP

Configuring support for MacIP for the router involves the configuration of a few specific commands and requires several general configuration prerequisites. In general, MacIP-related configuration steps are as follows:

Step 1: Establish a MacIP server for a specific zone.

Step 2: Specify at least one *dynamic* or *static* resource address assignment statement for each server.

These are the only steps necessary. However, in order for MacIP to function properly, several conditions must be met:

- AppleTalk routing must be enabled on at least one interface.
- IP routing must be enabled on at least one interface.
- The MacIP zone name configured must be associated with a configured or *seeded* zone name.
- Any IP address specified in configuring a given MacIP server using an **appletalk macip** configuration statement must be *aliasable* to a specific IP interface on the router. Since the router is acting as a proxy for MacIP clients, it is not acceptable to use an IP address to which the router ARP module is unable to respond.

Enabling MacIP Servers

Use the **appletalk macip server** global configuration command to establish a new MacIP server. The command syntax is:

appletalk macip server *ip-address* **zone** *server-zone*
no appletalk macip server *ip-address* **zone** *server-zone*

Only one MacIP server can be configured per AppleTalk zone. A server is not registered via NBP until at least one MacIP resource is configured.

The **no appletalk macip** command shuts down all active MacIP services. If entered with the keyword **server**, a specific *ip-address* and a specific *server-zone*, the particular server statement (if one exists) will be shut down and eliminated from the configuration.

Example:

The following example illustrates specification of a MacIP server on interface Ethernet 0 in AppleTalk zone Engineering. Also provided are some related IP and AppleTalk configuration commands.

```
!This global statement specifies server address and zone:
appletalk macip server 131.108.1.27 zone Engineering
!
!These statements assign the address and subnet mask for Ethernet0:
interface ethernet 0
ip address 131.108.1.27 255.255.255.0
!
!These statements specify the AppleTalk zone Engineering for Ethernet0:
appletalk routing
!
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Engineering
```

Note: Multiple MacIP servers can be configured for a router, but only one can be assigned to a particular zone and only one IP interface is assigned to each MacIP server. In general, the IP address assigned with this command must be *aliasable* to an existing IP interface address. For implementation simplicity, Cisco suggests that the address specified in this command match an existing IP interface address.

Specifying Addresses for Dynamic MacIP Clients

Use the **appletalk macip dynamic** global configuration command to allocate a single IP address or a range of IP addresses to be assigned to *dynamic* MacIP clients by the MacIP server serving zone *server-zone*. Dynamic clients are those who accept any IP address assignment within the dynamic range specified. The command syntax is:

appletalk macip dynamic *ip-address* [*ip-address*] **zone** *server-zone*
no appletalk macip dynamic *ip-address* [*ip-address*] **zone** *server-zone*

The **no appletalk macip** command disables all MacIP services. If entered with the keyword **dynamic**, a specific *ip-address* range, and a specific *server-zone*, then the particular dynamic address assignment statement is eliminated from the configuration.

Example:

The following example illustrates MacIP support for dynamically addressed MacIP clients with dynamically allocated IP addresses in the range of 131.108.1.28 to 131.108.1.44.

```
!This global statement specifies server address and zone:
appletalk macip server 131.108.1.27 zone Engineering
!
!This global statement specifies dynamically-addressed clients:
appletalk macip dynamic 131.108.1.28 131.108.1.44 zone Engineering
```

```
!
!These statements assign the address and subnet mask for Ethernet0:
interface ethernet 0
ip address 131.108.1.27 255.255.255.0
!
!These statements specify the AppleTalk zone Engineering for Ethernet0:
appletalk routing
!
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Engineering
```

Specifying Addresses for Static MacIP Clients

Use the **appletalk macip static** global configuration command to define a range of addresses to be made available to MacIP clients who have reserved an invariant IP address. The server keeps track of these address for routing and informational purposes. The command syntax is:

appletalk macip static *ip-address* [*ip-address*] **zone** *server-zone*
no appletalk macip static *ip-address* [*ip-address*] **zone** *server-zone*

The **no appletalk macip** command shuts down all running MacIP services. If entered with the keyword **static**, a specific *ip-address* and a specific *server-zone*, the particular static address assignment statement (if one exists) will be eliminated from the configuration.

Note: In general, Cisco recommends that you do not use fragmented address ranges if possible in configuring ranges for MacIP. However, in some cases it might be unavoidable. In such cases, use the **appletalk macip static** command to assign a specific address or address range.

Example:

The following example illustrates MacIP support for MacIP clients with statically allocated IP addresses. Addresses range from 131.108.1.50 to 131.108.1.66. Nodes have the specific addresses 131.108.1.81, 131.108.1.92 and 131.108.1.101.

```
!This global statement specifies server address and zone:
appletalk macip server 131.108.1.27 zone Engineering
!
!These global statements specify statically-addressed clients:
appletalk macip static 131.108.1.50 131.108.1.66 zone Engineering
appletalk macip static 131.108.1.81 zone Engineering
appletalk macip static 131.108.1.92 zone Engineering
appletalk macip static 131.108.1.101 zone Engineering
!
!These statements assign the address and subnet mask for Ethernet0:
interface ethernet 0
ip address 131.108.1.27 255.255.255.0
!
```

```
!These statements specify the AppleTalk zone Engineering for Ethernet0:
appletalk routing
!
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Engineering
```

MacIP Configuration and Address Assignment Considerations

Remember, the following configuration when setting up MacIP routing:

- Static and dynamic resource statements are cumulative. The administrator can specify as many as necessary. It is desirable to specify a single all-inclusive range if possible, as opposed to several adjacent ranges; that is, 131.108.121.1 to 131.108.121.10 as opposed to 131.108.121.1 to 131.108.121.5 and 131.108.121.6 to 131.108.121.10.
- Overlapping resource ranges (that is, 131.108.121.1-131.108.121.5 and 131.108.121.5-131.108.121.10) are *not* allowed. If it is necessary to change a range in a running server, use the negative form of the resource address assignment statement (such as **no appletalk macip dynamic**) to delete the original range, followed by the corrected range statement.
- It is always possible to add resources to a running server, as long as the new range does not overlap with one of the old ranges.

AppleTalk Access and Distribution Lists

Cisco's AppleTalk access lists provide network security by permitting or denying certain packets access to a network interface. Cisco's AppleTalk access lists are applicable to *zones* or *networks*; they may not be used for specific nodes.

An *access list* is a list of AppleTalk network numbers or zones kept by the Cisco router to control access to or from specific networks or zones for a number of services.

Cisco's AppleTalk access-list capability supports four basic access control filter applications. Each uses AppleTalk access lists and can be defined on a per-interface basis. The supported filters are as follows:

- Packet filtering (zones are ignored)
- Routing data generation
- Routing data acceptance (zones are ignored)
- *Get-zone-list* handling (networks are ignored)

The subsequent discussions focus on the following topics:

- Alternative access-list implementation approaches
- Command descriptions and definitions
- Configuration examples illustrating use of AppleTalk access lists

AppleTalk Access Control Methods

Cisco supports two general classes of AppleTalk *access control lists* (ACLs):

- True AppleTalk-style ACLs (based on AppleTalk *zones*)
- IP-style access lists (supported with prior software releases and based on *network number*)

The chief advantage of AppleTalk-style ACLs is that they allow you to define access regardless of the existing network topology or any changes in future topologies—ostensibly because they are based on zones. A *zone ACL* is effectively a dynamic list of network numbers. The user specifies a zone name. The effect is as if the user had specified all of the network numbers belonging to that zone.

Zone-Based AppleTalk Access Control

AppleTalk-style ACLs regulate the internetwork using zone names. Since zone names are the only network-level abstraction that users can access, this is the ideal control point. Cisco routers permit routing to be controlled using zone names—stated either explicitly or using generalized argument keywords. AppleTalk ACLs thus allow for simplified network management and greater flexibility in terms of adding segments with a reduced reconfiguration requirement for the router.

Note: Network entries and zone entries may be combined in a single list and have a cumulative effect. Network filtering is performed first and then zone filtering is applied to the result. However, for optimal performance, access lists should not include both zones and numeric network entities.

Network Number-Based AppleTalk Access Control

In general, Cisco does not recommend specification of IP-style ACLs to control network access. However, it can be done by creating access lists based on network number. Such controls are not optimal, because they ignore the logical mapping provided by AppleTalk zones. Since partially obscuring a zone is not a defined facility, the list of network numbers must be configured in each secured router. When networks are added to a zone, those networks must be enumerated at each secure router.

Add to this administrative overhead the fact that anyone can add network segments (for example, finance gets a LaserWriter and installs a Cayman Gatorbox—thereby creating a new network segment), and the potential for confusion and misconfiguration is significant.

Nonetheless, Cisco routers do allow you create IP-style ACLs. In particular, this may useful in permitting the definition of network lists that control the disposition of networks that overlap, are contained by or exactly match, a specific network range.

Note: One class of problem addressed by the use of network-number based access lists involves the potential assignment of conflicting (same) network numbers to different networks. An access control list can be used restrict the network numbers and zones that a department can advertize, thereby limiting advertisement to an authorized set of networks. In general, zone-based ACLs are not enough in this application.

Creating AppleTalk Access Lists

To use access lists, two sets of commands are needed. The first set defines an access list. The second defines how the access list is to be used; in other words, these commands associate an access list with a specific interface or specify that it is to be used as a routing filter. To define an access list, use the **access-list** global configuration command. This command has several optional formats and supports *extended* AppleTalk addressing, as follows:

access-list *list* {**permit**|**deny network**} *network*
no access-list *list* **permit**|**deny network** *network*

access-list *list* {**permit**|**deny**} **cable-range** *start-end*
no access-list *list* {**permit**|**deny**} **cable-range** *start-end*

access-list *list* {**permit**|**deny**} **includes** *start-end*
no access-list *list* {**permit**|**deny**} **includes** *start-end*

access-list *list* {**permit**|**deny**} **within** *start-end*
no access-list *list* {**permit**|**deny**} **within** *start-end*

access-list *list* {**permit**|**deny**} **zone** *zonename*
no access-list *list* {**permit**|**deny**} **zone** *zonename*

no access-list *list*

access-list *list* {**permit**|**deny**} **additional-zones**

access-list *list* {**permit**|**deny**} **other-access**

The argument *list* is an integer from 600 to 699.

The argument *network* is an AppleTalk network number.

The argument *start-end* is a cable range value (decimal number from 1 to 65279, inclusive). The starting network number should be less than or equal to the ending network number.

The argument *zonename* specifies the name of the zone for the connected AppleTalk network. The argument *zonename* may include special characters from the Apple Macintosh character set. To include a special character, insert a colon and two uppercase hexadecimal characters.

Additional **permit** and **deny** conditions may be added to the list by issuing further **access-list** commands for that list.

Note: Unlike the access lists of other protocols, the ordering of conditions is unimportant. As a result, no network entry may overlap any other entry in a single list.

Use the **no access-list** command with the *list* number only to remove an entire access list from the configuration. Specify the optional arguments to remove a particular clause.

The following descriptions define the **access-list** command variations (specified above) and outline the use and behavior of each:

access-list *list* {**permit**|**deny**} **network** *network*
no access-list *list* {**permit**|**deny**} **network** *network*

Specifies AppleTalk access control list (ACL) for a single network number. Affects matching non-extended networks. This rule is used when an exact match is made. Ranges of zero (in other words, same starting and ending number) do not match a network entry with that specific number.

Note: The software versions predating SW Release 9.0, the **access-list** command did not include the **network** keyword. If entered into a configuration or found in a boot file, this prior form of the command is transformed into the new form outlined above. The pre-9.0 syntax is accepted as if the **network** keyword were present. Similarly, the network number -1 (previously used to specify *any* network) is accepted as representing the **other-access** keyword. The forms documented here are generated for the **write** command.

---

access-list *list* {**permit**|**deny**} **cable-range** *start-end*
no access-list *list* {**permit**|**deny**} **cable-range** *start-end*

Specifies ACL for an extended network. The **access-list** command applies to extended networks with the matching starting and ending numbers. This rule is used when an exact match is to be made.

access-list *list* {**permit**|**deny**} **includes** *start-end*
no access-list *list* {**permit**|**deny**} **includes** *start-end*

Specifies ACL for any network, extended or nonextended, that overlaps any part of the range of values *start* through *end*, inclusive.

access-list *list* {**permit**|**deny**} **within** *start-end*
no access-list *list* {**permit**|**deny**} **within** *start-end*

Specifies ACL for any network, extended or nonextended, whose range of network numbers is included entirely within start through end, inclusive.

access-list *list* {**permit**|**deny**} **zone** *zonename*
no access-list *list* {**permit**|**deny**} **zone** *zonename*

Specifies ACL that applies to any network that has the specified *zonename* in its zone list.

access-list *list* {**permit**|**deny**} **additional-zones**

Specifies ACL used for zone-related checks to specify the default action for zones that were not enumerated. If not specified, the default is to deny additional zones. A **no** version is not applicable to this variation of the **access-list** command.

access-list *list* {**permit**|**deny**} **other-access**

ACL used as the default for any case that was not enumerated. If not specified, the default is to deny other access. A **no** version is not applicable to this variation of the **access-list** command.

Example:

This example illustrates removal of all clauses associated with AppleTalk access list 610 and the removal the specific zone named Subhumans in AppleTalk access list 620.

```
no access-list 610
no access-list 620 zone Subhumans
```

Assigning an Access List to an Interface

A *packet filter*, specified via the **appletalk access-group** interface subcommand, prevents any packets from being sent out an interface if the source network has access denied. Once assigned, no packet that fails the **appletalk access-list** command will go out on that interface. The full syntax of this command follows:

appletalk access-group *access-list-number*
no appletalk access-group *access-list-number*

The argument *access-list-number* is the number of a predefined access list in the range of 600 to 699, inclusive. If an undefined access list is used, the rule defaults to **permit**. If a zone does not match any rule in the list, it is denied, unless permitted via the **other-access** option of the **access-list** global configuration command.Use the **no appletalk access-group** command to remove the list from the interface.

Access lists applied using the **access-group** interface subcommand must permit the network number of the outgoing interface.

The EXEC command **show appletalk traffic** displays the number of packets dropped because of access control. Refer to the section on "Monitoring the AppleTalk Network" later in this chapter for more information. See the section "Filtering Networks Sent Out in Updates" for an example of the use of this command.

When defining access lists for an interface, all networks within a zone should be governed by the same access control.

Note: A zone-specific ACL will have no effect when applied using the **appletalk access-group** interface subcommand.

Filtering Networks Received in Updates

Use the **appletalk distribute-list in** interface subcommand to filter routing updates received from other routers over the specified interface. The full syntax for this command follows:

appletalk distribute-list *access-list-number* **in**
no appletalk distribute-list *access-list-number* **in**

The argument *access-list-number* is the number of an Appletalk access list defined by a set of **access-list** commands.

When AppleTalk routing updates are received on the specified interface, each network number and range in the update is checked against the access list. Only network numbers and ranges that are permitted by the access list are inserted into the router's Appletalk routing table.

Use the **no appletalk distribute-list** *access-list-number* **in** command to remove this function.

Note: Incoming routing data is checked using the network entries of the ACL.When using an AppleTalk input distribution list, the assigned access control list should not contain any zone entries since the resulting behavior is undefined.

Example:

These commands cause any mention of network 10 to be ignored in routing updates arriving via Ethernet 3.

```
access-list 601 deny network 10
access-list 601 permit other-access
!
interface ethernet 3
appletalk distribute-list 601 in
```

Filtering Networks Sent Out in Updates

Use the interface configuration subcommand **appletalk distribute-list out** to filter routing data generated from zones or networks. The full syntax of this command follows.

appletalk distribute-list *access-list-number* **out**
no appletalk distribute-list *access-list-number* **out**

The argument *access-list-number* is the number of an AppleTalk access list defined by a set of **access-list** commands. If an undefined access list is used, the rule defaults to permit. If a zone does not match any rule in the list, it is denied unless permitted via the **other-access** option of the **access-list** global configuration command.

A *distribution list* is a list of AppleTalk access list numbers kept by the Cisco router which controls whether the network numbers specified by the access list are processed during the reception or transmission of routing updates. A distribution list will not prevent packets destined for a specified network number from being accepted; it will only prevent the route to the specified network from appearing in neighboring routers' AppleTalk routing tables.

Note: After performing network filtering, each network is checked for possible omission due to zone filtering.

Use the **no appletalk distribute-list** *access-list-number* **out** command to remove this function.

Example:

These commands prevent routing updates sent on Ethernet 0 from mentioning any networks in zone Admin. The **appletalk access-group** command prevents packets from being sent out the interface.

```
access-list 601 deny zone Admin
access-list 601 permit other-access
interface Ethernet 0
appletalk distribute-list 601 out
appletalk access-group 601
```

Defining Get-Zone-List Filters

Use the **appletalk getzonelist-filter** interface subcommand to modify zone-list replies. The syntax for this command is:

appletalk getzonelist-filter *access-list-number*
no appletalk getzonelist-filter *access-list-number*

The argument *access-list-number* must be in the range of 600 to 699, inclusive. If an undefined access list is used, the rule defaults to **permit**. If a zone does not match any rule in the list, it is denied, unless permitted via the **additional-zones** option of the **access-list** global configuration command.

Note: Using a get-zone-list (GZL) filter is not a complete replacement for anonymous network numbers. In order to prevent users from seeing a zone, all routers must implement the GZL filter. If there are any non-Cisco routers on the network, the GZL filter will not have a consistent effect.

Use the **no appletalk getzonelist** *access-list-number* command to remove this function.

A GZL request is used by the Macintosh's Chooser to find a list of zones from which the user can select services. Any router on the same network as the Macintosh may respond with a GZL reply. The GZL filter is used to cause the router to omit certain zones in its reply. Note that this filter only changes the list of zones presented to the user. It does not change the router's behavior in routing packets or in processing routing updates. You must use the other filters to achieve those goals.

All routers on a given network should filter get-zone replies identically. Otherwise Macintoshes will present different zone lists to the user depending upon which router responds to the request. Inconsistent filters can result in zones appearing and disappearing every few seconds when the user remains in the Chooser. If there are non-Cisco routers on the network, the command **appletalk getzonelist-filter** is not likely to be useful unless the non-Cisco routers have a similar feature.

Note: The reply to a get-zone list request is also filtered by any **appletalk distribute-list out** filter in effect for the interface involved. You only need to define an **appletalk getzonelist-filter** command if you want additional filtering to be applied to GZL replies. This filter is rarely needed except to eliminate zones that do not contain user services.

Permitting Partial Zones

If any network of a zone is denied, then the zone is also denied unless the global AppleTalk global configuration command **appletalk permit-partial-zones** is enabled. The command syntax is:

appletalk permit-partial-zones
no appletalk permit-partial-zones

The default is for **appletalk permit-partial-zones** to be *disabled*.

Specifying the keyword sequence **permit-partial-zones** disables the default behavior where the complete zone is access controlled if any associated network is controlled. In other words, when a specific zone is partially obscured, other (visible) networks that are not subject to access control are propagated normally when **permit-partial-zones** is enabled.

The **no appletalk permit-partial-zones** version of this command disables this option, and restores the default condition where a complete zone is controlled if any associated network is controlled.

If **permit-partial-zones** is enabled, AppleTalk cannot maintain consistency for the nodes in the affected zones and the results are undefined. With this option enabled, an inconsistency is created for the zone and several assumptions made by some Appletalk protocols are no longer valid.

Note: This feature provides IP-style access control with similar functionality to the new AppleTalk-style access control lists. If enabled, the access control list behavior associated with prior software releases is restored. In addition, NBP protocol cannot ensure consistency and uniqueness of name bindings.

Requiring Specific Route Zones

Use the **appletalk require-route-zones** global configuration command to prevent *bogus* routes (possibly generated by a broken router or corrupt packet) from causing ZIP protocol storms. The command syntax is:

appletalk require-route-zones
no appletalk require-route-zones

The default is for **require-route-zones** to be *enabled*.

ZIP protocol storms can arise when corrupt routes are propagated and routers broadcast ZIP requests to determine the network/zone associations.

When **require-route-zones** is enabled, the router will not advertise a route to its neighboring routers until it has obtained the network/zone associations. This effectively limits the storms to a single network rather than the entire internet.

Use the **no appletalk require-route-zones** command to disable the **require-route zones** option and set the condition such that the router can advertise routes to its neighbors without having obtained the network-zone associations.

Disabling this feature enables the routing behavior associated with prior software releases; when enabled, this option *requires* that all networks have zone names prior to advertisement to neighbors.

As an alternative to disabling this option, *empty* zones can be filtered from the list presented to users while the pertinent networks can be associated with a zone name for monitoring purposes. This is done with the **appletalk getzonelist-filter** interface subcommand described earlier in this chapter.

The *user* zone lists may be configured to vary from interface to interface, but this is discouraged since AppleTalk users expect to have the same user zone lists at any end-node in the internet. This kind of filtering does not prevent explicit access via programmatic methods, but should be considered a user optimization whereby unused zones are suppressed. Other forms of AppleTalk access control lists should be used to actually *secure* a zone or network.

Controlling AppleTalk Names Displayed

Two global configuration commands control the Cisco router's name-display feature: **appletalk lookup-type** and **appletalk name-lookup-interval**. The names and services specified with the **appletalk lookup-type** command are held in a lookup cache and displayed using **show appletalk name-cache** EXEC command.

Note: Node numbers do not change very frequently because each device keeps track of the last node number it was assigned. Typically, node numbers only change if a device is shutdown for an extended period of time or is moved to a new network segment.

Setting Service Types Cached

Use the **appletalk lookup-type** global configuration command to specify services listed in **show appletalk nbp** and **show appletalk name-cache** EXEC command display. The command syntax is:

appletalk lookup-type *serviceType*
no appletalk lookup-type [*serviceType*]

The argument *serviceType* is the specific AppleTalk service.

- **ciscoRouter**—Listed in **show appletalk nbp** display per port
- **SNMP Agent**—Listed in **show appletalk nbp** display per zone if and only if Apple's SNMP-over-DDP is enabled

Note: If AppleTalk routing is enabled, enabling SNMP will automatically enable SNMP-over-DDP. Also, if you include this entry with your list of **appletalk lookup-type** commands, enter it *as is*—with the space between SNMP and Agent.

- **IPGATEWAY**—Active MacIP server names
- **IPADDRESS**—Active MacIP server addresses

Other common service types, each of which can be specified in an **appletalk lookup type** command:

- **AppleRouter**—Apple internet router
- **GatorBox**—Cayman's LocalTalk gateway
- **Workstation**—System 7 Macintosh (also, the machine type is defined as an additional name by defining both, it is possible to easily identify all user nodes)
- **FastPath**—Shiva's LocalTalk gateway
- **systemRouter**—Cisco's OEM router name
- **SNMP**—Identifies node supporting IP SNMP (only done by some vendors and now considered obsolete)

Note: These non-Cisco services only appear in a display generated using the **show appletalk name-cache** command. Also, if a neighboring router is not one of Cisco's routers, or is running pre-9.0 software, it is possible the router will be unable to determine the name of the neighbor. This is normal behavior and there is no workaround.

As many **appletalk lookup-type** commands may be issued as desired. The service type **ciscoRouter** is the only type initially enabled; it cannot be disabled. Cisco routers generate requests to the network segments to which they are directly connected so the name cache does not contain entries for *all* the selected services in a zone—only those which are *directly connected*.

The interval can be set to be as frequent or infrequent as desired.

Entries are deleted after several interval periods expire without the entry being refreshed. At each interval, a single request is sent via each interfaces that have valid addresses. Refer to the description of **appletalk name-lookup-interval** that immediately follows this command for a discussion of setting the name lookup interval.

The command **no appletalk lookup-type** can be used with or without the *serviceType* argument. Using the argument specifies exclusion of a specific service type from the name cache. Prevent all names (except those relating to Cisco routers) from being cached by using the **no** version of this command without the argument *serviceType*.

Note: When specifying a service-type name with this command, spaces are valid (such as **SNMP Agent**). However, do not use leading or trailing spaces when entering these names.

Example:

The following simple example illustrates the use of this command to specify various services to be listed in **show appletalk name-cache** display.

```
appletalk lookup GatorBox
! In addition to ciscoRouter, check for GatorBox services
appletalk lookup AppleRoute
! and Apple Internet Routers
appletalk lookup IPGATEWAY
! and MacIP servers...
appletalk lookup Workstation
! Not generally needed for any but the most inquiring minds.
! Note ciscoRouter automatically is listed
```

Setting Service Name Lookup Interval

Use the **appletalk name-lookup-interval** global configuration command to set the interval between service pollings by the router on its AppleTalk interfaces. The command syntax is:

appletalk name-lookup-interval *intInSeconds*
no appletalk name-lookup-interval

The argument *intInSeconds* is the interval in seconds between NBP lookup pollings. The Cisco router does not query the entire zone, but instead polls only the connected networks to reduce overhead.

An interval of 0 (zero), the default, disables the **appletalk lookup-type** feature.All polling for services is suspended. By reentering a nonzero, positive integer value for *intInSeconds*, the **appletalk lookup-type** specifications in the active configuration are reinstated. A value of 0 (zero) is equivalent to **no appletalk name-lookup-interval**. You cannot disable lookup of **ciscoRouter**.

Note: After disabling this parameter, the name cache is purged at the next global configuration run.

There is no limit on this value, but the recommended values are 300 (five minutes) and 1200 (20 minutes). The smaller the interval, the more packets are generated to handle the names.

Example:

The following example illustrates setting the lookup interval:

```
appletalk name-lookup-interval 1200
! Lookup names once every 20 minutes
```

AppleTalk Configuration Examples

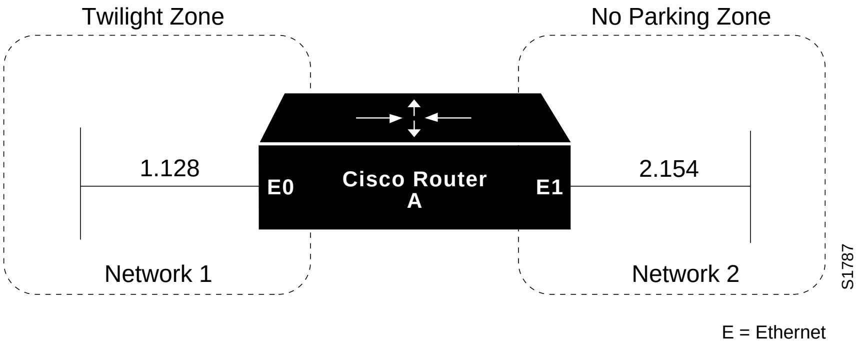
The following examples illustrate a variety AppleTalk configurations:

- Configuring nonextended AppleTalk networks routing between two Ethernets
- Setting up a transitional configuration, where routing is occurring between nonextended and extended AppleTalk networks
- Configuring nonextended AppleTalk networks routing over X.25
- Creating a simple configuration for an extended AppleTalk network
- Setting up extended AppleTalk networks routing over HDLC
- Initializing SNMP configuration for AppleTalk
- Configuring IPTalk
- Building and using AppleTalk access lists

Nonextended AppleTalk Routing Between Two Ethernets

This example configuration illustrates how to configure routing between two Ethernets. Ethernet 0 is on network 1, at node 128. Ethernet 1 is on network 2, at node 154. The two networks are in the Twilight and No Parking zones, respectively. See Figure 1-6 for an illustration.

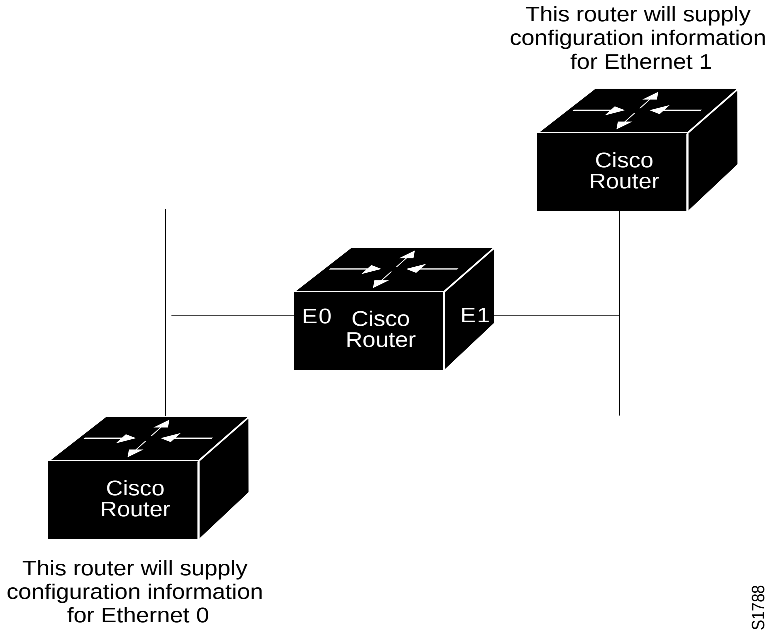
Figure 1-6 Nonextended AppleTalk Routing Between Two Ethernet Networks



```
appletalk routing
!
interface ethernet 0
appletalk address 1.128
appletalk zone Twilight
!
interface ethernet 1
appletalk address 2.154
appletalk zone No Parking
```

The next example is a variation of the above configuration. It differs in that it has other seed routers on both networks to provide the zone and network number information. In this way, the Cisco router discovers the information dynamically. Refer to Figure 1-7 for an illustration.

Figure 1-7 Routing Between Seed Routers


This router will supply configuration information for Ethernet 1
Cisco Router
E0
Cisco Router
E1
Cisco Router
This router will supply configuration information for Ethernet 0
S1788

```
appletalk routing
!
interface ethernet 0
appletalk address 0.0
!
interface ethernet 1
appletalk address 0.0
```

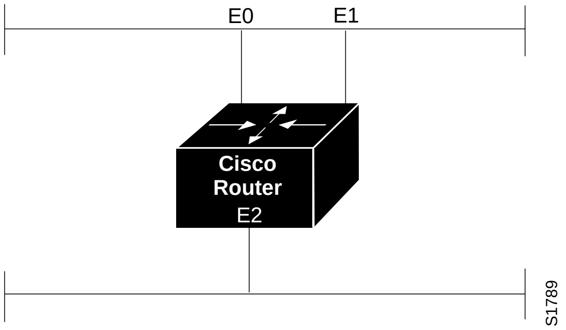
Configuring Transition Mode

The Cisco router may be used to route between extended and nonextended AppleTalk networks that exist on the same cable. Other vendors have coined the term *transition mode* for this type of routing.

To do this on the Cisco router, you must have two ports connected to the same physical cable. One port is configured as a nonextended AppleTalk network, and the other as an extended AppleTalk network.

Both ports must have unique network numbers because you are actually routing between two separate AppleTalk networks: an extended and a nonextended network. Figure 1-8 shows an example of the topology and configuration of such connection.

Figure 1-8 Transition Mode Topology and Configuration



```
interface ethernet 0
appletalk cable-range 2-2
appletalk zone No Parking
!
interface ethernet 1
appletalk address 3.128
appletalk zone Twilight
!
interface ethernet 2
appletalk cable-range 4-4
appletalk zone Do Not Enter
```

Note: Networks 2-2 and 4-4 in the above example have a cable range of one and a single zone in their zone lists. This must be true to maintain compatibility with the nonextended network, Network 3.

Nonextended AppleTalk Routing over X.25

The configuration of X.25 networks is similar to that for HDLC encapsulation. However, you must completely and explicitly configure all network and node numbers in an X.25 environment. Note that all AppleTalk nodes within an X.25 network must be configured with the same AppleTalk network number.

X.25 configuration for AppleTalk involves mapping AppleTalk addresses to X.121 addresses, executed with the X.25 configuration subcommand **x25 map** (see the section "Configuring the Datagram Transport on Commercial X.25 Networks" in the "Configuring Packet-Switched Software" chapter).

Each time a packet is sent to a particular AppleTalk address, that address is looked up in the X.25 map table in order to match it to an X.25 address. The packet is encapsulated in X.25 frames and sent to the X.25 node which is its destination.

The receiving node reassembles the X.25 frames if necessary, then strips the packet of X.25 framing information so that the original AppleTalk datagram can be processed.

In the configuration commands that follow, the keyword **broadcast** (as used at the end of the **x25 map** commands) has the following effect: whenever a broadcast packet is sent, assuming the broadcast flag is set, then each X.121 address specified will receive the broadcast. The X.25 protocol does not provide broadcasts; therefore, they must be simulated in this manner when using X.25 as a transport protocol for another protocol that requires broadcasts, such as AppleTalk.

If the X.121 address of the router on the far end of the X.25 network is 123456789012, and your local X.121 address is 210987654321, and the two routers are at AppleTalk addresses 7.63 and 7.25, you would configure these systems in the following way.

```
!Configuration for First Router
interface serial 0
appletalk address 7.25
appletalk zone Twilight
x25 map appletalk 7.63 123456789012 broadcast
!Configuration for Second Router
interface serial 0
appletalk address 7.63
appletalk zone Twilight
x25 map appletalk 7.25 210987654321 broadcast
```

In this example, a third router has the X.121 address 333444555666 and AppleTalk address *7.100*.

```
!Configuration for Third Router
interface serial 0
appletalk address 7.100
appletalk zone Twilight
x25 map appletalk 7.25 210987654321 broadcast
x25 map appletalk 7.63 123456789012 broadcast
```

With the addition of the third router, both the original routers need an additional **x25 map** entry:

```
x25 map appletalk 7.100 333444555666 broadcast
```

Note: X.25 may be configured only as a nonextended network using the **appletalk address** command. Logically, it is the same as a localtalk network since both are *always* non-extended networks.

Extended AppleTalk Routing Network

The following example illustrates how to configure an extended AppleTalk network.

This configuration defines the zones *Empty Guf* and *Underworld* from which the router and the nodes may choose to reside. The equal cable range numbers allow compatibility with nonextended AppleTalk networks.

```
appletalk routing
!
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Empty Guf
appletalk zone Underworld
```

Extended AppleTalk Routing over HDLC

AppleTalk's dynamic address assignment feature allows users and network managers to choose default network addresses.   The following example illustrates configuration of two ends of a serial line for routing of AppleTalk over HDLC. An example of the interface configuration for both ends of the serial line follows.

Example:

The following commands enable AppleTalk routing for interface serial 1. Assuming that a serial link is made between two different routers (both using interface serial 1), then the configuration can be the same for both ends of the connection.

```
interface serial 1
appletalk cable-range 1544-1544
appletalk zone Twilight
```

Configuring SNMP in AppleTalk Networks

For AppleTalk to enable SNMP-over-DDP, AppleTalk routing must be active before the SNMP configuration, otherwise the AppleTalk SNMP server will not be started. This is done correctly with the standard configuration handling.

However, problems can arise if AppleTalk is started manually when the SNMP server was previously configured for the router. The following example configuration sequence illustrates proper activation of SNMP and AppleTalk on a router.

Specification of the **snmp-server** global configuration commands must *follow* the **appletalk routing** global configuration commands and **interface** subcommand specifications.

Note: Refer to the chapter "Configuring the System" for information about configuration SNMP for the router. The **snmp-server** commands are global configuration commands.

Example SNMP Configuration for an AppleTalk Router:

The following example briefly illustrates the command sequence needed when starting AppleTalk routing and an SNMP server process on a router from the console.

```
no snmp-server
!
appletalk routing
appletalk event-logging
!
interface Ethernet 0
ip address 131.108.29.291 255.255.255.0
appletalk cable-range 29-29 29.180
appletalk zone Zombie
!
snmp-server community propellerhead RW
snmp-server trap-authentication
snmp server 131.108.2.160 propellerhead
```

Configuring IPTalk

IPTalk is AppleTalk *encapsulated* in IP datagrams. IPTalk is used to route across backbones and to communicate to applications on hosts that are unable to communicate via AppleTalk. The CAP is an example. The following discussion describes setting up UNIX-based systems and the Cisco router to use CAP IPTalk and other IPTalk implementations.

Note: If your system is a Sun or DEC ULTRIX system, it may be possible to directly run CAP in a mode that supports EtherTalk. In that case, your system looks like any other AppleTalk node and does not need any special IPTalk support. However, other UNIX systems for which EtherTalk support is not available in CAP must run CAP in a mode that depends upon IPTalk.

IPTalk Configuration Steps

The procedure that follows outlines the basic steps for setting up Cisco routers and UNIX hosts for operation using IPTalk implementations.

Note: The procedure that follows does not give full instructions on how to install CAP on the UNIX system. This discussion specifically addresses the required steps for setting up the UNIX-system's configuration file that defines addresses and other network information. Generally, this is the only file that relies on the Cisco address and configuration information. The rest of the setup for UNIX systems involves building the CAP software and setting up the UNIX startup scripts that make it run. These peripheral discussions are beyond the scope of this manual.

Step 1: Set up the Cisco routers for AppleTalk. The routers talk to each other and Apple products using more standard protocols, such as EtherTalk or TokenTalk. IPTalk is needed only on an interface that will communicate with a UNIX system. You must have AppleTalk routing enabled among all of the routers that are going to use IPTalk. This includes any routers in the middle that are required for them to be able to communicate with each other. Otherwise the UNIX systems cannot communicate with each other.

Step 2: Ensure that IP is enabled on the interface to be used to communicate with the UNIX system. Refer to the "Routing IP" and "IP Routing Protocols" chapters for more information about configuring IP. Since IPTalk is AppleTalk encapsulated in IP, IP must be enabled on the router *and* on the UNIX system. This interface must be on *the same subnet* as the UNIX system.

Step 3: Allocate an AppleTalk network number for IPTalk. A separate AppleTalk network number is needed for each IP subnet that is to run IPTalk. It is possible to have a number of UNIX machines on the same subnet. They all use the same AppleTalk network number for IPTalk. They must have their own individual node ids. It is possible for the same router to have IPTalk enabled on several interfaces. Each interface must have a different AppleTalk network number assigned for use by IPTalk, since each interface will be using a different IP subnet.

Step 4: Determine the CAP format of the AppleTalk network number. The CAP software is based on an old convention, which expresses AppleTalk network numbers as two octets (numbers from 0 to 255) separated by a dot. The Apple convention uses decimal numbers from 1 to 65279. Use the following formula to convert between the two:

CAP format: x.y
Apple format: d

- Converting from Apple to CAP: x = d/256; y = d%256
  ("/" represents truncating integer division; and % the remainder)
- Converting from CAP to Apple: d = x * 256 + y

Example:
Apple format: 14087; CAP format: 55.7

Step 5: Decide on a zone name for IPTalk. There are no special constraints on choice of zone name. The same zone name can be used for several networks. IPTalk and normal networks can be combined in the same zone if desired.

Step 6: Decide which UDP ports you are going to use for IPTalk. The default is to use ports beginning with 768. Thus, RTMP uses port 769, NBP port 770, and so on. These are the original ports hardcoded into older versions of CAP. The only problem with using them is that the port numbers are not officially assigned by the Internet's Network Information Center (NIC). Thus other applications could use them, possibly causing conflicts—although this is unlikely. The NIC has assigned a set of UDP ports beginning with 200. Beginning with CAP release 5.0, it became possible to configure CAP to use the officially allocated ports. If you do so, RTMP will use port 201, NBP port 202, and so on. If you decide to use the these or other ports, you must configure both CAP and the Cisco router to use the same ports.

Step 7: Enable IPTalk on each interface of the Cisco router as required. Here is an example:

```
appletalk routing
!
interface ethernet 0
ip address 128.6.7.22 255.255.255.0
! EtherTalk phase 2
appletalk cable 1792-1792 1792.22
appletalk zone MIS-Development
! IPTalk
appletalk iptalk 14087.0 MIS-UNIX
```

In this example, Ethernet 0 is configured to speak AppleTalk in two different ways:

- Via EtherTalk phase 2 using network number 1792 and zone MIS-Development
- Via IPTalk using network number 14087 and zone MIS-UNIX

Note: The node id is not specified (is left as 0) in the **appletalk iptalk** global configuration command. The IPTalk node id is chosen automatically, based on the IP address. It is normally the host number portion of the IP address. For example, with an IP address of 128.6.7.22 and a subnet mask of 255.255.255.0, the host number is 22. Thus, the IPTalk node id is 22. If the IP host number is larger than 255, the low-order 8 bits are used, although fewer than 8 bits may be available depending on the IP subnet mask. If the mask leaves fewer bits, the node number will be quietly truncated. Be sure to use a node address that is compatible with the subnet mask. In any event, there are likely to be problems using IPTalk with host numbers larger than 255.

If you choose to use the official UDP ports (those beginning with 200), use the following command configuration line in your configuration:

```
appletalk iptalk-baseport 200
```

Note: This line is not an interface command; it may go before or after the interface commands.

Step 8: Configure each UNIX host with the correct network number, zone name, and router.

As an example, here are the contents of */etc/atalk.local* from a UNIX system with IP address 128.6.7.26:

```
# IPTalk on net 128.6.7.0:
# mynet mynode myzone
55.7  26      MIS-UNIX
# bridgenet bridgenode bridgeIP
55.7  22      128.6.7.22
```

The first noncomment line defines the address of the UNIX system; the second line defines the Cisco router. In both cases, the first column is 55.7, which is the AppleTalk net number chosen for use by IPTalk (in CAP format). The second column is the AppleTalk node id, which must be the same as the IP host number. The third column is the zone name on the first line, and the IP address of the Cisco router on the second line.

Note that the following must agree:

- The first column in both lines must agree with the AppleTalk network number used in the **appletalk iptalk** configuration command. However in */etc/atalk.local* it must be in the CAP format, and in the configuration command it must be in the Apple format.
- The second column in both lines must agree with the IP host address of the corresponding system (the UNIX machine for the first line, the Cisco router for the second line).
- The third column in the first line must agree with the zone name used in the **appletalk iptalk** configuration command.
- The third column in the second line must agree with the IP address of the Cisco router.

Step 9: Make sure that your CAP software is using the same UDP port numbers as the Cisco router. Currently, CAP's default is the same as Cisco's (in other words, port numbers beginning with 768). If you want to use this default, you do not need to take no further action with regard to this step. However, to use the official UDP port numbers, make sure that you have used the following global configuration command (described above):

```
appletalk iptalk-baseport 200
```

Step 10: On the UNIX system, add the following lines to the file */etc/services*:

```
at-rtmp       201/udp
```

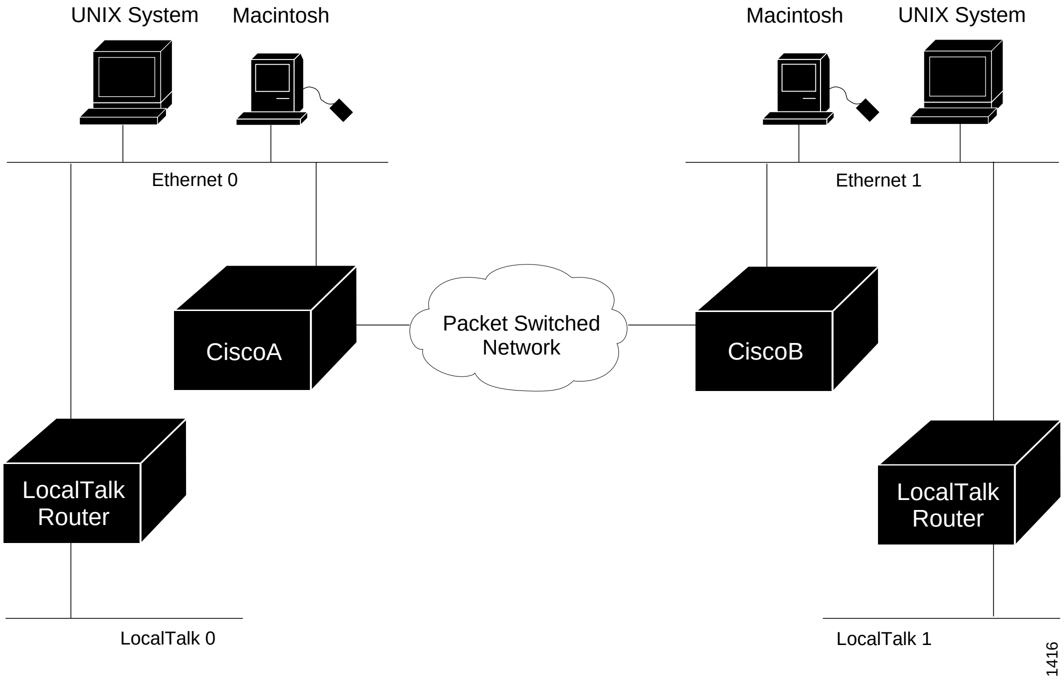
```
at-nbp        202/udp
at-3          203/udp
at-echo       204/udp
at-5          205/udp
at-zis        206/udp
at-7          207/udp
at-8          208/udp
```

If you are using Network Information Services (NIS), also commonly referred to as *yellow pages*, remember to do a *make* in */var/yp* after changing */etc/services*. If you are using the default ports, those starting with 768, you do not need the **appletalk iptalk-baseport** command, nor do you need to modify the file */etc/services*.

IPTalk Configuration Note

The installation instructions for CAP refer to KIP gateways and to the file *atalkatab.* If you use Cisco's IPTalk support, the file *atalkatab* is neither necessary nor desirable. Cisco's IPTalk support assumes that you wish all wide-area AppleTalk routing to be done using the normal AppleTalk routing protocols. KIP and *atalkatab* is based on a alternative routing strategy, where AppleTalk packets are passed around the Internet using IP routing. It is possible to use both strategies at the same time; however, the interaction of the two routing techniques is not well defined.

Figure 1-9 IPTalk Configuration Example



If you have routers from other vendors that support *atalkatab*, you should disable *atalkatab* support on them, in order to avoid this mixed routing. The installation instructions that come with some of these products encourage you to use *atalkatab* for complex networks. When you are using Cisco routers, this is not necessary. The Cisco implementation of IPTalk integrates IPTalk into the normal AppleTalk network routing. Consider the example network diagram illustrated in Figure 1-9.

In this example, CiscoA and CiscoB must enable both standard AppleTalk (EtherTalk) and IPTalk on the Ethernets shown. They will use EtherTalk to communicate with the LocalTalk Routers and Macintoshes, and IPTalk to communicate with the UNIX systems. The LocalTalk Routers should also have both EtherTalk and IPTalk enabled. IPTalk should be configured with *atalkatab* disabled. The LocalTalk Routers will use IPTalk to communicate with the UNIX systems adjacent to them, and EtherTalk to communicate with the rest of the AppleTalk network.

If you did not enable IPTalk on the LocalTalk Routers, the system will still work. However systems on LocalTalk that wished to communicate with the nearby UNIX system would have to go through the Cisco router. This creates an unnecessary extra hop, since the LocalTalk Routers can speak IPTalk directly to the UNIX system.

Note: In this configuration, all traffic between systems on the left and the right transit via the Cisco routers using AppleTalk routing. If *atalkatab* support is enabled on the LocalTalk Routers, this would establish a *hidden* path between them, unknown to the more standard AppleTalk routing protocols. In a large network, this can result in traffic taking inexplicable routes.

AppleTalk Access List Configuration Examples

The **access-list** command examples that follow illustrate several AppleTalk access-list filter variations and contrast different approaches to access control list application.

Basic Access List Example

The following is a compilation of typical **access-list** statements:

```
access-list 601 permit zone ZoneA
access-list 601 permit zone ZoneB
access-list 601 deny zone ZoneD
access-list 601 deny additional-zones
access-list 601 permit network 55
access-list 601 permit network 500
access-list 601 permit cable-range 900-950
access-list 601 deny includes 970-990
access-list 601 permit within 991-995
access-list 601 deny other-access
```

These separate statements combine to establish access list number 601 with the following characteristics:

- Permits routing tuples to any network that has Zone A specified in its zone list
- Permits routing tuples to any network that has Zone B specified in its zone list
- Denies routing tuples to any network that has Zone D specified in its zone list
- Blocks routing tuples to any zone not specifically enumerated
- Permits routing tuples for nonextended network 55 and allows routing of packets destined for network 55
- Permits routing tuples for nonextended network 500 and allows routing of packets destined for network 500
- Permits routing tuples for the cable range 900-950, and allows routing of AppleTalk packets destined for any network in that range
- Denies any routing tuple that has a starting or ending network number within the range 970 and 990 inclusive, and prevents the routing of AppleTalk packets destined for any network in that range
- Permits any routing tuple that has both starting and ending network numbers within the range of 991 and 995 inclusive, and allows routing of AppleTalk packets destined to any network in that range
- Denies ACL routing tuples for any case that was not enumerated

Note: When applying an access control list such as this with the various interface subcommands (**access-group**, **distribute-list**, or **getzonelist-filter**) if an undefined access list is used, it defaults to **permit**; if a condition being tested is not handled by the specified access list, the router denies access by default.

To illustrate how the router tests incoming routing information against its access lists and interface specifications, consider the following test responses to detected conditions. Assume that the access list clauses for 601 are applied to a particular router interface. The following outcomes result from hypothetical tests:

- If the interface access control specification is testing a zone name of Zone C, no test is successful, so the **additional-zones** setting, deny for the example and by default, is the result.
- If the interface access control specification is testing a zone name of Zone B, the result is permit due to an explicit match.
- If the interface access control specification is testing a zone name of Zone D, the result is deny due to an explicit match.

- If the interface access control specification is testing a cable range of 55-55, the result is the **other-access** setting, deny for the example and by default. A cable-range of 55 does not match a network number of 55 for the purposes of distribution list testing. However, if this list is used as an access-group, 55 does match.
- If the interface access control specification is testing a cable-range of 972-980, the result is to deny by explicit match.

Distribution list filtering operates on *exact* matches when making comparisons. The comparison is between a incoming routing tuple (which considers 55 and 55-55 to be different) and the condition defined in the access control list.

The process for accepting or rejecting routing information when applying distribution lists can be further defined with some illustrative examples.

Table 1-2 through Table 1-4 list the results associated with a specific test condition. If the outcome value is *true*, the condition passes the access list specification and the **distribute-list** interface subcommand specification is applied.

Table 1-2 **Test Condition #1:** Routing tuple of 55

Example Access List Options Configured in Router	Outcome of Test
access-list 601 permit network 55	True
access-list 601 permit cable 55-55	False
access-list 601 permit includes 55-55	True
access-list 601 permit within 55-55	True

Table 1-3 **Test Condition #2:** Testing routing tuple of 55-55

Example Access List Options Configured in Router	Outcome of Test
access-list 601 permit network 55	False
access-list 601 permit cable 55-55	True
access-list 601 permit includes 55-55	True
access-list 601 permit within 55-55	True

Table 1-4 **Test Condition #3:** Testing routing tuple of 55-60

Example Access List Options Configured in Router	Outcome of Test
access-list 601 permit network 50	False
access-list 601 permit network 55	False
access-list 601 permit cable 50-55	False
access-list 601 permit cable 50-50	False
access-list 601 permit cable 50-60	True
access-list 601 permit includes 50-55	True

Example Access List Options Configured in Router	Outcome of Test
access-list 601 permit includes 55-55	True
access-list 601 permit includes 50-60	True
access-list 601 permit within 50-55	False
access-list 601 permit within 55-55	False
access-list 601 permit within 50-60	True

For the **access-groups** interface subcommand specifications, used to control *packet flow*, the destination network number is used and all clauses are tested as if the test condition (**network**, **cable**, **includes**, or **within**) is actually **includes**. So, for the destination network of 55, all of the above test outcomes are *True* (when tested with **access-groups**) *except* for `network 50` and `cable 50-50`.

Note: For any set of values, no condition may overlap within the *same* access list. For this purpose, 50-50 and 50, are considered overlapping. However, access control lists used for different purposes on the same interface, may contain entries which overlap in the different lists.

Comparison of Alternative Segmentation Solutions

With the flexibility allowed by Cisco's access list implementation, determining the optimal method to segment an AppleTalk environment using access control lists can be unclear.The following scenario and configuration examples illustrate how two solutions can solve a particular problem and discusses the inherent advantages of using AppleTalk-style access lists.

Consider a situation where a company's management wants to permit customers to have direct access to several corporate file servers. Access to all devices in zones named MIS and Corporate is to be permitted, but other access discouraged since unconstrained permission might facilitate unauthorized access to sensitive engineering file servers. The solution: create appropriate access control lists to enforce access policies.

The environment for this internet is comprised of the following networks and zones:

- Zone: Engineering. Network numbers/cable ranges: 69-69, 3, 4160-4160, 15
- Zone: MIS. Network numbers/cable ranges: 666-777
- Zone: Corporate. Network numbers/cable ranges: 70-70, 55, 51004, 4262-4262
- Zone: World. Network numbers/cable ranges:88-88, 9, 9000-49999 (multiple networks exist in this range)

The router named Gatekeeper is placed between the World zone and the various company-specific zones. There can be an arbitrary number of routers on either side of Gatekeeper. An Ethernet backbone exists on each side which connects these other routers to Gatekeeper. (E0 is the World backbone and E1 is the Corporate backbone).

For the purposes of this configuration, assume Gatekeeper is the only router which needs any access list configuration. There are two solutions depending on the level of security desired.

A minimal configuration might be as follows (the Engineering zone is secured, but all other zones are publicly accessible):

```
int ether 0
appletalk distrib 601 out
appletalk access 601
!
access-list 601 deny zone Engineering
access-list 601 permit additional-zones
access-list 601 permit other-access
```

A more comprehensive configuration might be as follows (Corporate and MIS zones are public; all other zones are secured):

```
int ether 0
appletalk distrib 601 out
appletalk access 601
!
access-list 601 permit zone Corporate
access-list 601 permit zone MIS
access-list 601 deny additional-zones
access-list 601 deny other-access
```

Both configurations satisfy the basic goal of isolating the engineering servers, but the second example will continue to be secure when additional zones are added in the future.

Get-zone-list (GZL) Configuration Example

The following is an example of a get-zone-list (GZL) access filter implementation. In addition to the basic configuration commands, this example also provides the following:

- A discussion of the sequence of testing and route elimination associated with this filtering mechanism
- Lists the resulting information that will be included in the GZL

A GZL reply, per AppleTalk, contains a list of all zones. This can be modified by access lists to be a list of all zones which are associated with visible network entities and not explicitly excluded by an access list. The following configuration defines an access list that is used to modify the GZL, for interface Ethernet 0:

```
access-list 601 permit zone A
access-list 601 permit zone B
access-list 601 deny net 300
access-list 601 deny includes 1-100
access-list 601 permit other-access
access-list 601 permit zone D
access-list 601 deny additional-zones

access-list 602 permit zone A
access-list 602 permit zone B
access-list 602 deny additional-zones
```

```
int ether 0
appletalk distrib 601 out
appletalk getzonelist 602
```

The discussion that follows focuses on outlining the process of removing unwanted entries from an initial AppleTalk *zone/network association table.*

For the purposes of illustration, Table 1-5 matches the access list entries with arbitrary rule numbers. These rule numbers are then used to describe the process of route elimination employed by the AppleTalk access control mechanism.

Table 1-5 GZL Filter Example Access List Rules

Access List Entry	Rule Number
access-list 601 permit zone A	1
access-list 601 permit zone B	2
access-list 601 deny net 300	3
access-list 601 deny includes 1-100	4
access-list 601 permit other-access	5
access-list 601 permit zone D	6
access-list 601 deny additional-zones	7
access-list 602 permit zone A	8
access-list 602 permit zone B	9
access-list 602 deny additional-zones	10

Table 1-5 depicts a hypothetical initial state for an AppleTalk zone-network association table. This get-zone-list is then modified with the tests described in the following discussion.

Table 1-6 Initial Zone-Network Association Table

Network Number	Zone Name	Zone-Network Association
1-5	A	a1
98-102	A	a2
300	B	a3
400	C	a4
401	A	a5
402-402	D, B	a5

The first test applied to the router is to eliminate networks which are covered by access lists. The following network-zone associations are eliminated from the get-zone-list table:

- a1 and a2 are eliminated as a result of rule 4 (listed in Table 1-5).
- a3 is eliminated by rule 3.

Table 1-5 lists the network-zone associations that remain after the first test is completed on the initial table (elimination of networks from table per access list specification).

Table 1-7 Zone-Network Association Table After Access List Applied to Network

Network Number	Zone Name	Zone-Network Association
400	C	a4
401	A	a5
402-402	D,B	a6

The next test is the application of zone filtering using the distribution list. Network-zone association *a4* is eliminated from the get-zone-list table as a result of applying rule 7, since no other zone rule applied.

Table 1-5 lists the network-zone associations that remain after the distribution list test is completed on the list in Table 1-5 (elimination of network 400 per **deny additional-zones** access list specification).

Table 1-8 Zone-Network Association Table After Distribution List Test

Network Number	Zone Name	Zone-Network Association
401	A	a5
402-402	D, B	a6

Finally, zone filtering is applied via the **appletalk getzonelist-filter**. Network-zone association *a6* is eliminated from the get-zone-list table as a result of rule 10 zone D failed to meet any other zone rule.

Thus, the get-zone-list table will contain only a single entry (of those found in the initial table)—zone A.

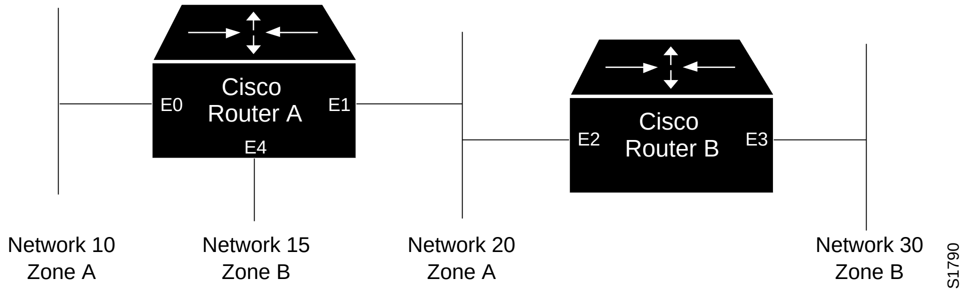
Partial Zone Advertisement Configuration Example

Figure 1-10 illustrates a situation where you might want to allow for the partial advertisement of a particular zone.

Figure 1-10 Example Topology of Partially Obscured Zone

Assume that RouterB includes a distribution filter (applied with the **appletalk distribute-list** interface subcommand) on the interface Ethernet 3 that excludes Network 10. The associated commands might look like this

```
access-list 612 deny network 10
```



```
int eth 3
appletalk distribute-list 612 out
appletalk distribute-list 612 in
```

For Network 30, normal (default) behavior would be for Network 10 and Network 20 to be eliminated from any routing updates sent, although Network 15 would be included in routing updates (same zone as Network 30). Using the **appletalk permit-partial-zones** configuration command has the following effects:

- If permit-partial-zones is enabled (**appletalk permit-partial-zones**), the routing updates exclude Network 10, but *includes* Network 15 and Network 20.
- If permit-partial-zones is disabled (**no appletalk permit-partial-zones**), the routing updates exclude both Network 10 and Network 20, still include Network 15. This is generally considered the preferred behavior, and is the default.

Table 1-9 provides an overview of the associations between the networks illustrated in Figure 1-10. Table 1-10 details the effects of enabling and disabling partial-zone advertisement with the **permit-partial-zones** command.

Table 1-9 Zone and Interface Associations for Partial Zone Advertisement Example

	Network 10	Network 15	Network 20	Network 30
Zone	A	B	A	B
Interface(s)	Ethernet 0	Ethernet 4	Ethernet 1 Ethernet 2	Ethernet 3

Table 1-10 Partial-zone Advertisement Control on Network 30

Command Condition	Network 10	Network 15	Network 20	Network 30
Enabled	Not Advertised on Network 30	Advertised on Network 30	Advertised on Network 30	—
Disabled	Not Advertised on Network 30	Advertised on Network 30	Not Advertised on Network 30	—

Hiding and Sharing Access to Resources with Access Lists

The following examples illustrate the use of AppleTalk access lists to manage access to certain resources.

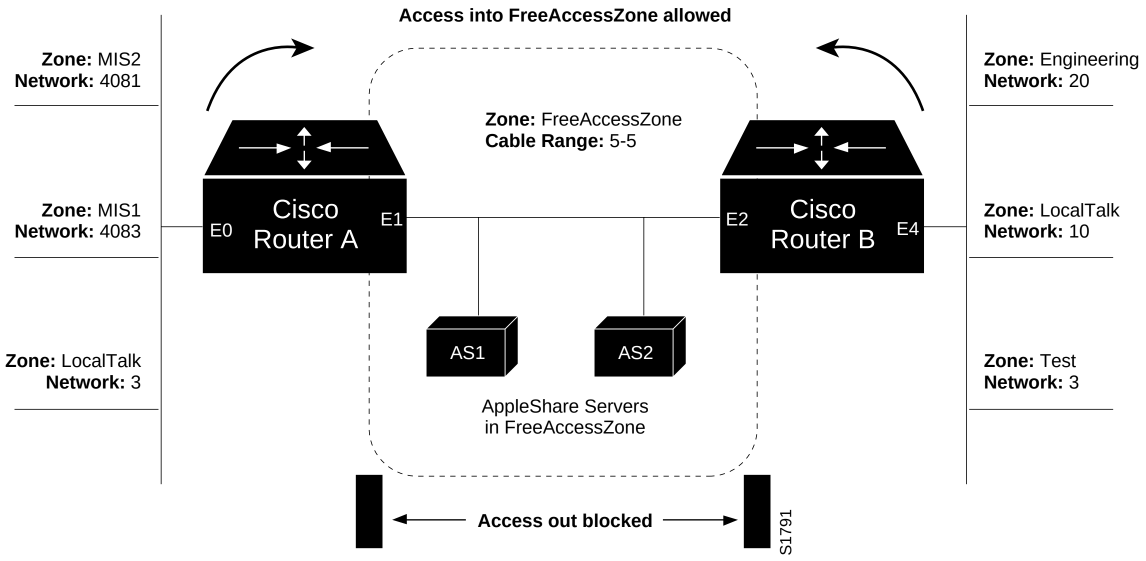
Establishing Free Access to Common AppleShare Servers

Consider a situation in which you wish to provide access to several AppleShare servers on a a network directly connected to two routers, but want to restrict cross-access among other networks that are connected to these routers. Figure 1-11 illustrates an environment that reflects this situation. The configuration example and associated discussion describe how access lists can be used to provide control.

The following configuration listing provides the configuration for Router A and Router B in Figure 1-11. Only interface Ethernet 1 (E1 on Router A) and interface Ethernet 2 (E2 on Router B) are configured in order to provide the control described here.

In this example, the goal of network administrators is to allow all users on the various networks connected to both Router A and Router B to be able to access the AppleShare servers AS1 and AS2 in zone FreeAccessZone. A second requirement is to block cross-access through this zone. In other words, users in zones MIS1, MIS2, and LocalTalk (connected to Router A, interface E0) are not allowed to have access to any of the resources on networks connected to interface E4, on Router B. Similarly, users in zones Engineering, Test, and LocalTalk (connected to RouterB, interface E4) are not allowed to have access to any of the resources on networks connected to interface E0, on Router A.

Figure 1-11 Diagram Illustrating Controlling Access to Common AppleTalk Network



Note: Although there are networks that share the same number on interfaces E0 and E4, and zones that have the same name, none have the same network number and zone specification (except FreeAccessZone). The two routers do *not* broadcast information about these networks through FreeAccessZone. The routers only broadcast the cable range 5-5. As configured, FreeAccessZone only sees itself. However, since no other limitations have been placed on advertisements, the FreeAccessZone range of 5-5 propagates out to the networks attached to E0 (RouterA) and E4 (RouterB), and thus resources in FreeAccessZone are made accessible to users on all those networks.

Router A Configuration:

```
! Global configuration specification of access list 601
access-list 601 permit cable 5-5
access-list 601 deny other-access
!
interface ethernet 1
appletalk cable-range 5-5
appletalk zone FreeAccessZone
appletalk distribute-list 601 out
appletalk distribute-list 601 in
```

Router B Configuration:

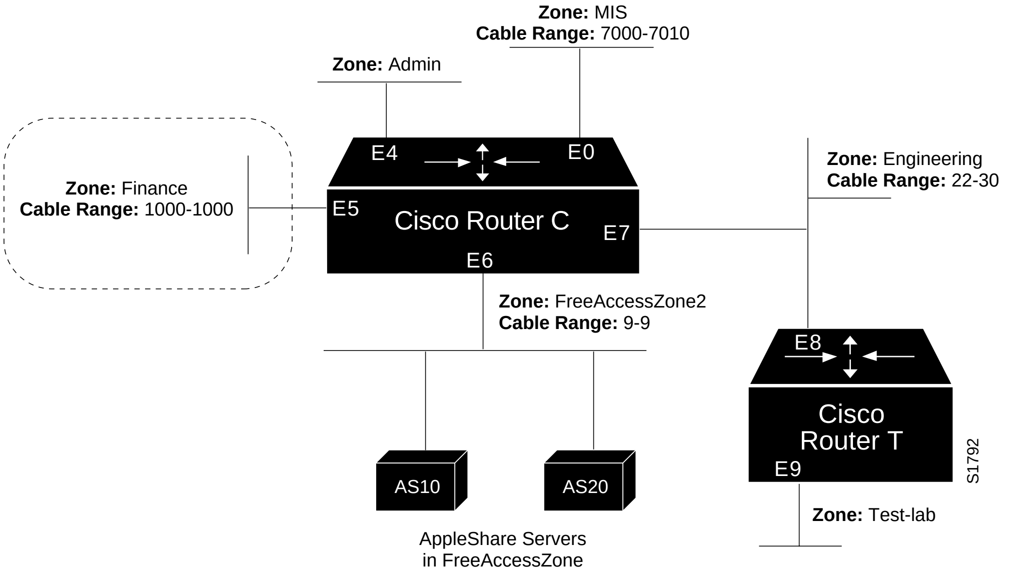
```
! Global configuration specification of access list 601
! (access lists are identical to RouterA)
access-list 601 permit cable 5-5
access-list 601 deny other-access
!
interface ethernet 2
```

```
! Specifications for Ethernet 2 on RouterB are same
! as specifications for Ethernet 1 on RouterA
appletalk cable-range 5-5
appletalk zone FreeAccessZone
appletalk distribute-list 601 out
appletalk distribute-list 601 in
```

Restricting Resource Availability with Access Lists

In the preceding example, shared-resource access was granted to all users in the various AppleTalk zones connected to the two routers. At the same time, access between resources on either side of the common zone was completely denied. There may be instances where a greater degree of control is required—possibly where resources in some zones are to be allowed access to resources in certain other zones, but denied access to other specific zones. Figure 1-12,the accompanying discussion, and the configuration command examples that follow the figure, illustrate such a situation.

Figure 1-12 Controlling Resource Access Among Multiple AppleTalk Zones



The following guidelines are the administrative objectives for this example:

- Users in zones Engineering (E7) and MIS (E0) are to be allowed free access to each other.
- All users in all zones are to be allowed access to FreeAccessZone2 (E6).
- No users in any zone, with the exception of users in Finance, are to be allowed access to resources in Finance.

Router C Access List Configuration:

The preceding specifications require three access lists. The configuration commands associated with each are listed below:

```
! Global configuration specification of access list 601:
access-list 601 permit cable 9-9
access-list 601 deny other-access
! Global configuration specification of access list 610:
access-list 610 permit zone Finance
access-list 610 permit zone FreeAccessZone2
access-list 610 deny additional-zones
! Global configuration specification of access list 620:
access-list 620 deny cable 11-12
access-list 620 permit cable 100-150
access-list 620 deny other-access
access-list 620 permit cable 200-200
```

The effects of these access lists can be briefly defined as follows:

- Access list 601 mirrors the access control list provided in the preceding example (illustrated in Figure 1-11). It is intended to be used to allow access to resources on FreeAccessZone2.
- Access list 610 is intended to be used to control access in and out of the zone Finance.
- Access list 620 is intended to be used to accommodate the requirement to allow users networks 100-150 and 200-200 to mutually access network resources.

The specification of the **access-list** global configuration command is only the first half of the access control process. The second half of the process is to then apply the access lists to interfaces using the various **appletalk** interface subcommands. The assignment of access lists to specific interfaces for this example follows.

Router C Interface Ethernet 0 Configuration:

Interface Ethernet 0 is associated with MIS. The configuration command listing is as follows:

```
interface ethernet 0
appletalk cable-range 7000-7010
appletalk zone MIS
appletalk distribute-list 620 out
appletalk distribute-list 620 in
```

By specifying access list 620 using **distribute-list** interface subcommands, the following results:

- FreeAccessZone2 is advertised into MIS.
- Advertisements of Finance are blocked.
- Advertisements between Engineering and MIS are allowed.

Router C Interface Ethernet 5 Configuration:

Interface Ethernet 5 is associated with Finance. This zone requires some specific controls. The configuration command listing for Ethernet 5 is as follows:

```
interface ethernet 5
```

```
appletalk cable-range 1000-1000
appletalk zone Finance
appletalk distribute-list 610 out
appletalk access-group 610
```

The configuration for Ethernet 5 requires using both a **distribute-list out** interface subcommand and an **access-group** subcommand. Using these commands has the following effects:

- With the **distribute-list out** subcommand, Finance is limited to accessing Finance and FreeAccessZone2 only.
- The **access-group** subcommand filters packet traffic, thus it blocks access to any devices in *Finance* from outside of this zone.

Router C Interface Ethernet 6 Configuration:

FreeAccessZone2 is on interface Ethernet 6. Since users in all zones are to have access to resources in this zone, the configuration is as follows:

```
interface ethernet 6
appletalk cable 9-9
appletalk zone FreeAccessZone2
appletalk distribute-list 601 out
appletalk distribute-list 601 in
```

Router C Interface Ethernet 7 Configuration:

Ethernet 7 has a configuration that mirrors the configuration for Ethernet 0, since the users in zones MIS and Engineering are to have mutual resource access. The configuration would be as follows:

```
interface ethernet 7
appletalk cable-range 22-30
appletalk zone Engineering
appletalk distribute-list 620 out
appletalk distribute-list 620 in
```

Implicit Configuration of Zones Admin and Test-Lab

Conspicuously omitted from this configuration listing are any specific configuration listings pertaining to the Test-Lab (on Router T interface E9, and connected to Router C via E7) and Admin (on Router C interface E4). These zones are omitted because there are no requirements listed in the original objectives relating to them. Some access control is implicitly handled with the assignment of the stated access lists:

- Users in zone Admin can see the Finance zone, but cannot see resources in that zone. However, as for all zones, resources in FreeAccessZone2 are available, but none of the users in any of the other zones can access resources in Admin.
- In the absence of the assignment of access lists on Router T, users in Test-Lab can access the resources in FreeAccessZone2 and zone Engineering. With the exception of Engineering, no other zones can access resources in Test-Lab.

Monitoring the AppleTalk Network

Use the EXEC **show** commands described in this section to obtain displays of activity on the AppleTalk network.

Displaying AppleTalk Access List Specifications

Use the **show appletalk access-lists** EXEC command to display conditions specified in AppleTalk access list configurations. The following is a sample output from the associated configuration commands:

```
AppleTalk access list 601:
        permit zone ZoneA
        permit zone ZoneB
        deny additional-zones
        permit network 55
        permit network 500
        permit cable-range 900-950
        deny includes 970-990
        permit within 991-995
        deny other-access
```

Displaying the Adjacent Routes

The **show appletalk adjacent-routes** EXEC command results in a display of routes that are directly connected or one hop away. When an AppleTalk internet has more then 600 networks, this command gives administrators a quick synopsis of the local environment.

You can use information provided in this display to determine what local routes are missing or misconfigured so that appropriate action can be taken.

To show the routing table for adjacent routes, use the **show appletalk adjacent-routes** EXEC command:

show appletalk adjacent-routes

Following is a sample display for an extended AppleTalk network:

```
Codes: R - RTMP derived, C - connected, 67 routes in internet

R Net 29-29 [1/G] via gatekeeper, 0 sec, Ethernet0, zone Engineering
C Net 2501-2501 directly connected, Ethernet1, no zone set
C Net 4160-4160 directly connected, Ethernet0, zone Low End SW Lab
C Net 4172-4172 directly connected, TokenRing0, zone Low End SW Lab
R Net 6160 [1/G] via urk, 0 sec, TokenRing0, zone Low End SW Lab
```

Displaying the ARP Cache

To display the AppleTalk ARP cache, use the following EXEC command:

show appletalk arp

This command displays the contents of the AARP cache. AARP establishes correspondences between network addresses and LAN hardware addresses (Ethernet addresses). A record of each correspondence is kept in a cache for a predetermined amount of time and then discarded. Following is sample output. Table 1-11 describes the fields seen.

```
Protocol  Address          Age (min)     Hardware Addr   Type      Interface
AppleTalk 4172.30                  -     0000.3080.84ab  SNAP      TokenRing0
AppleTalk 4160.19                 94     0000.0c00.0082  SNAP      Ethernet0
AppleTalk 4160.21                 94     0000.0c00.d8db  SNAP      Ethernet0
AppleTalk 2501.117                 -     0000.0c00.d8de  SNAP      Ethernet1
AppleTalk 4172.224               206     0000.3080.8453  SNAP      TokenRing0
AppleTalk 4160.150                 -     0000.0c00.d8dd  SNAP      Ethernet0
```

Table 1-11 Show IP Arp Field Displays

Field	Description
Protocol	Protocol for network address in the Address field
Address	The network address that corresponds to Hardware Addr
Age (min)	Age, in minutes, of the cache entry; entries are purged once they reach four hours (240 minutes) old
Hardware Addr	LAN hardware address that corresponds to network address

Type	Type of ARP (Address Resolution Protocol):
	ARPA = Ethernet-type ARP
	SNAP = RFC 1042 ARP

Displaying the Fast-Switching Cache

Use the **show appletalk cache** command with the extended AppleTalk networks to display the current fast-switching cache. Enter this command at the EXEC prompt:

show appletalk cache

This display includes the current cache version number and all entries (valid or not). Valid entries are identified by an asterisk (*) in the first column.

Conditions that invalidate the fast-switching cache are as follows:

- Route deleted but not marked bad (and has been used)
- A route that has gone bad (and has been used)
- When you replace a route with a new metric (and it was used)
- When a neighbor transitions from suspect to bad
- When a node address in the AARP cache changes hardware address
- When a hardware address changes node address
- When the AARP cache gets flushed
- When an AARP entry is deleted
- When the following configuration commands are entered:
    - After a **no appletalk routing** command

- — After an **appletalk route-cache** command
- — After an AppleTalk **access-list** command

- When the encapsulation for the line changes
- When a port leaves or enters Operational state

Following is a sample display of the **show appletalk cache** command:

```
AppleTalk Routing Cache, * = active entry, cache version is 227
 Destination Interface MAC Header
*       29.0 Ethernet0 00000C00008200000C00D8DD
*   1544.000 Ethernet1 AA000400013400000C000E8C809B84BE02
*     33.000 Ethernet1 AA000400013400000C000E8C809B84BE02
```

Displaying Global AppleTalk Information

The EXEC command **show appletalk global** displays information about the AppleTalk internetwork and specific parameters for the router. The command has this syntax:

show appletalk global

Following is a sample display:

```
AppleTalk global information:
  Internet is compatible with older, AT Phase1, routers.
  There are 67 routes in the internet.
  There are 25 zones defined.
  All significant events will be logged.
  ZIP resends queries every 10 seconds.
  RTMP updates are sent every 10 seconds.
  RTMP entries are considered BAD after 20 seconds.
  RTMP entries are discarded after 60 seconds.
  AARP probe retransmit count: 10, interval: 200.
  AARP request retransmit count: 5, interval: 1000.
  DDP datagrams will be checksummed.
  RTMP datagrams will be strictly checked.
  RTMP routes may not be propogated without zones.
  Alternate node address format will not be displayed.
  Access control of any networks of a zone hides the zone.
  Names of local servers will be queried every 60 seconds.
  Lookups will be generated for server types:
        appleRouter, Workstation, GatorBox
```

Displaying AppleTalk Interface Information

The **show appletalk interface** command displays AppleTalk-specific interface information. Enter this command at the EXEC prompt.

show appletalk interface [*interface*]

The argument *interface* specifies an interface name and number to display a specific interface.

This information displayed by this command includes the extended AppleTalk cable ranges and the current interface mode (the network verification/discovery mode, for example).

Sample displays of the **show appletalk interface** command follow.

Nonextended AppleTalk—Normal Operation:

```
Ethernet 1 is up, line protocol is up
  AppleTalk address is 666.128, Valid
  AppleTalk zone is Underworld
```

Extended AppleTalk—Normal Operation:

Depending on the configuration of the global configuration commands **appletalk lookup-type** and **appletalk name-lookup-interval,** a node name can appear in this display (in addition to the node address). For instance, in the example display output below, the node name urk is listed:

```
TokenRing 0 is up, line protocol is up
  AppleTalk cable range is 4172-4172
  AppleTalk address is 4172.30, Valid
  AppleTalk zone is "Low End SW Lab"
  AppleTalk port configuration provided by 4172.224 (urk)
  AppleTalk discarded 117 packets due to output errors
  AppleTalk discovery mode is enabled
  AppleTalk route cache is not supported by hardware
```

Extended AppleTalk—Verification Mode:

```
Ethernet 1 is up, line protocol is up
  AppleTalk routing disabled, Verifying port configuration
  AppleTalk cable range is 666-666
  AppleTalk address is 666.128, Valid
  AppleTalk zone is Underworld
```

Extended AppleTalk—Configuration Error:

```
Ethernet 0 is up, line protocol is up
  AppleTalk routing disabled, Port configuration error
  AppleTalk cable range is 70-70
  AppleTalk address is 70.128, Bad
  AppleTalk zone is Empty Guf
```

When you enter the EXEC command **show appletalk interface** with the *interface* argument, the display looks like this:

```
Ethernet 0 is up, line protocol is up
  AppleTalk cable range is 69-69
  AppleTalk address is 69.105, Valid
  AppleTalk zone is "Empty Guf"
  AppleTalk port configuration verified by 69.163
  AppleTalk discarded 3149 packets due to input errors
  AppleTalk discarded 71 packets due to output errors
  AppleTalk route cache is enabled
```

If AppleTalk routing is disabled on an interface, the display looks like this:

```
Ethernet 1 is up, line protocol is up
  AppleTalk protocol processing disabled
```

Displaying MacIP Status

Two **show** EXEC commands provide information concerning MacIP processes:

- **show appletalk macip-servers**
- **show appletalk macip-clients**

MacIP traffic statistics are displayed via the **show apple traffic** and the **show apple macip-traffic** commands.

Each **show** command is described in the brief sections that follow.

Monitoring MacIP Servers

Use the **show appletalk macip-servers** command to get information concerning the status of the servers for a router. The command syntax is:

show appletalk macip-servers

The following is a sample output for this command:

```
MACIP SERVER 1, IP 131.108.199.221,  ZONE 'S/W Test Lab' STATE is
server_upResource #1 DYNAMIC 131.108.199.1-131.108.199.10, 1/10 IP in use
Resource #2 STATIC 131.108.199.11-131.108.199.20, 0/10 IP in use
```

A listing is provided for each MacIP server on the router. The following information is listed:

- MACIP SERVER—The number (arbitrarily assigned) of the MacIP server.
- IP—The IP address specified for the MacIP server.
- ZONE—The AppleTalk server zone specified in the **appletalk macip server** command.
- STATE—The state of the server, as described in Table 1-12.
- Resource—Lists resource specifications as defined in the **appletalk macip dynamic** and **appletalk macip static** configuration statements. Specifies whether the resource address is assigned dynamically or statically; identifies the IP address range associated with the resource specification; and indicates the number of active MacIP clients.

This display is very useful in determining the status of your MacIP configuration. In particular, the STATE field can help identify problems in you AppleTalk environment. The following are hints for using this information:

- If the STATE remains at resource_wait, it is possible that no resources have been assigned (with either the **appletalk macip dynamic** or **appletalk macip static** commands).

- If the STATE remains at zone_wait, it is possible that an incorrect *server-zone* is specified in the **appletalk macip server** command.

In addition, **show macip-servers** can be used along with **show appletalk interface** to identify AppleTalk problems:

Step 1: First, you can determine the state of the MacIP server using **show macip-servers**. If the STATE field persistently indicates an anomalous status (something besides server_up, such as resource_wait or zone_wait), then a problem exists.

Step 2: Next, execute a **show appletalk interface** command; with this command you can determine the status of AppleTalk routing and the specific interface itself.

Step 3: If the protocol and interface are up, then inspect the MacIP configuration statements for IP address and zone specification inconsistencies.

The output of the **show macip-servers** command provides an indication of the current state of each configured MacIP server. Each server operates according to a simple finite-state machine table, described in Table 1-12.

Table 1-12 MacIP State Table

State	Event	New State	Notes
initial	ADD_SERVER	resource_wait	“server” configured
resource_wait	TIMEOUT	resource_wait	wait for resources
resource_wait	ADD_RESOURCE	zone_wait	wait for zone seeding
zone_wait	ZONE_SEEDED	server_start	register server
zone_wait	TIMEOUT	zone_wait	wait until seeded
server_start	START_OK	reg_wait	wait for server reg
server_start	START_FAIL	del_server	couldn't start (config err?)
reg_wait	REG_OK	server_up	registration successful
reg_wait	REG_FAIL	del_server	reg. failed (duplicate IP?)
reg_wait	TIMEOUT	reg_wait	wait until register
server_up	TIMEOUT	send_confirms	NBP confirm all clients
send_confirms	CONFIRM_OK	server_up	
send_confirms	ZONE_DOWN	zone_wait	zone or IP interface down, restart
*	ADD_RESOURCE	*	ignore, except resource_wait
*	DEL_SERVER	del_server	“no server” statement (HALT)
*	DEL_RESOURCE	ck_resource	ignore
ck_resource	YES_RESOURCES	*	return to previous state
ck_resource	NO_RESOURCES	resource_wait	shutdown, wait for resources

The following are descriptions of the state functions:

- **initial**—All servers begin here.
- **resource_wait**—Wait until a client range has been configured for the server.
- **zone_wait**—Wait until the configured AppleTalk zone name for the server is up. Warning: the server will remain in this state if no such zone has been configured or if AppleTalk routing is not enabled.
- **server_start**—Register configured IPADDRESS, and register as IPGATEWAY. Open ATP socket to listen for IP address assignment requests. Send NBP lookup requests for existing IPADDRESSes, and automatically add clients with addresses within one of the configured client ranges.
- **server_up**—Server has registered. Enable routing to client ranges. Respond to IP address assignment requests.

- **send_confirms**—Send NBP confirm tickles active clients every minute. Delete clients that have not responded within the last five minutes. Check IP and AppleTalk interfaces used by MacIP server. If down or reconfigured, restart server.
- **del_server**—All servers end here. Deregister NBP names, purge all clients and deallocate server resources.
- **ck_resource**—Make sure there is at least one client range available. If not, deregister NBP names and return to resource_wait state.
- *****—If in first column, represents "any" state. if in second column, represents a return to state from which a * state was called.

Monitoring MacIP Clients

Use the **show appletalk macip-clients** command to get information concerning the status of the known clients. The command syntax is:

show appletalk macip-clients

The **show macip-clients** command displays the IP and DDP address of all MacIP clients, and the last time the client responded to a NBP confirm request.

Clients are deleted after five minutes of not responding to NBP confirm requests on their allocated IP address.

The following is a sample output for this command:

```
131.108.199.1@[27001n,69a,72s] 45 secs    'S/W Test Lab'
```

The resulting display lists all known MacIP clients by IP address. Bracketed information includes the AppleTalk DDP address of the registered entity (network, node address, and socket number), followed by the time since the last NBP confirmation and name of the zone to which this particular MacIP client is attached.

Monitoring MacIP Traffic

Use the **show appletalk traffic** command to get information concerning the status of the MacIP traffic. The command syntax is:

show apple traffic

An IP alias is established for each MacIP client, and for the IP address of the MacIP server, if it does not match an existing IP interface address. The client aliases can be viewed with the **show ip aliases** command (described in the chapter "Routing IP").

Use the **show apple macip-traffic** command to obtain a detailed breakdown of MacIP traffic that is sent through a gateway from AppleTalk to IP through Cisco routers. The output from this command is different from the output of the **show apple traffic** command, which shows normal AppleTalk traffic generated, received, or routed by Cisco routers. The command syntax is as follows:

show apple macip-traffic

Displaying Nearby NBP Services

Use the **show appletalk name-cache** EXEC command to display list of NBP services of nearby routers or other devices that support NBP. The syntax is:

show appletalk name-cache

Note: The **show appletalk name-cache** command can be authorized by the administrator to display any AppleTalk services of interest in local zones, whereas the **show appletalk nbp** command is used to show services registered by the router.

This is sample output for the **show appletalk name-cache** command**:**

```
AppleTalk Name Cache:
  Net Adr Skt Name                    Type           Zone
 4160  19 254 gatekeeper              ciscoRouter    Low End SW Lab
 4160  21 254 bill                    ciscoRouter    Low End SW Lab
 4160 150 254 pag.Ethernet0           ciscoRouter    Low End SW Lab
 4172  30 254 pag.TokenRing0          ciscoRouter    Low End SW Lab
 4172 224 254 urk                     ciscoRouter    Low End SW Lab
 6160  69 254 urk                     ciscoRouter    Low End SW Lab
```

This information is held in the NBP name cache.

Support for names allows administrators to easily identify and determine the status of any associated device. This can be important in AppleTalk internetworks where node numbers are dynamically generated.

Note: The routers listed in this display (except pag) are running software images that predate Cisco SW Release 9.0 —which accounts for name differences in this display. Non-Cisco routers will also have a naming format that does not include an appended interface name. The interface (`ethernet0`) included in the derived name `pag.ethernet0` in this display refers to the router pag's view of the world—*not* the local router's view. They may be, but are not necessarily, the same. This feature allows you to determine the routers and their connected interface which is providing routing for any given AppleTalk network.

Displaying NBP Services Registered by Cisco Routers

Use the **show appletalk nbp** EXEC command to display the NBP name registration table. The command syntax is:

show appletalk nbp

The following is a sample output:

```
Net  Adr Skt Name                    Type            Zone
4160 211 254 pag.Ethernet0           ciscoRouter     Low End SW Lab
4160 211   8 pag                     SNMP Agent      Low End SW Lab
4172  84 254 pag.TokenRing0          ciscoRouter     LES Tokenring
4172  84   8 pag                     SNMP Agent      LES Tokenring
 200  75 254 myrouter.Ethernet1      ciscoRouter     Marketing    *
```

Note: The **show appletalk nbp** command is used to show services registered by the router, whereas the **show appletalk name-cache** command can be authorized by the administrator to display any AppleTalk services of interest in local zones.

In this display, the fields are as follows:

- **Net**—AppleTalk network number.
- **Adr**—Node address.
- **Skt**—DDP socket number.
- **Name**—Name of service.
- **Type**—Device type, varies depending on service. The Cisco service types are:
  - **ciscoRouter**—Listed in **show appletalk nbp** display per port
  - **SNMP Agent**—Listed in **show appletalk nbp** display per zone if and only if Apple's snmp-over-ddp is enabled.
  - **IPGATEWAY**—Active MacIP server names
  - **IPADDRESS**—Active MacIP server addresses

If an asterisk (*) appears in the far right margin, then the name registration is pending confirmation.

Displaying Neighboring Routers

The **show appletalk neighbor** EXEC command shows all AppleTalk routers that are directly connected to any of the networks to which this router is directly connected. It is from these neighboring routers, that this router obtains the AppleTalk network topology and most of the other information it needs to support the protocol. The command has this syntax:

show appletalk neighbor [*neighbor-address*]

The optional argument *neighbor-address* permits access to detailed statistics and other information associated with a particular neighbor.

For the command **show appletalk neighbor**, the display looks like this:

```
AppleTalk neighbors:
```

```
31.86, Ethernet8, uptime 133:28:06, last update 1 sec ago
81.82, Fddi0, uptime 266:11:44, last update 7 secs ago
81.81, Fddi0, uptime 267:30:28, last update 958334 secs ago
  Neighbor is down.
29.200, Ethernet3, uptime 263:45:50, last update 948440 secs ago
 Neighbor has restarted 2 times in 267:59:53.
  Neighbor is down.
81.80, Fddi0, uptime 268:00:08, last update 963617 secs ago
  Neighbor is down.
17.128, Ethernet2, uptime 133:26:43, last update 2 secs ago
 Neighbor has restarted 1 time in 268:00:21.
69.163, Ethernet0, uptime 268:00:25, last update 1 sec ago
```

Depending on the configuration of the global configuration commands **appletalk lookup-type** and **appletalk name-lookup-interval**, a node name can appear in this display (as well as a node address). For instance, in the example display output below, the node names urk, gatekeeper, and bill are listed:

```
AppleTalk neighbors:
  4172.224     urk      TokenRing0, uptime 63:35:42, 1 sec
        Neighbor has restarted 2 times in 125:16:47.
  4160.19      gatekeeper       Ethernet0, uptime 125:17:53, 1 sec
  4160.21      bill     Ethernet0, uptime 13:07:55, 5 secs
        Neighbor has restarted 5 times in 89:53:09.
```

For the command **show appletalk neighbor 69.163**, the display looks like this:

```
Neighbor 69.163, Ethernet0, uptime 268:00:52, last update 7 secs ago
We have sent queries for 299 nets via 214 packets.
Last query was sent 4061 secs ago.

We received 152 replies and 0 extended replies.
We have received queries for 14304 nets via 4835 packets.
We sent 157 replies and 28 extended replies.
We received 0 ZIP notifies.
We received 0 obsolete ZIP commands.
We received 4 miscellaneous ZIP commands.
We received 0 unrecognized ZIP commands.
We have received 92943 routing updates.
Of the 92943 valid updates, 1320 entries were invalid.
We received 1 routing update which were very late.
Last update had 0 extended and 2 nonextended routes.
Last update detail: 2 old
```

If the global configuration commands **appletalk lookup-type** and **appletalk-name-lookup interval** have been configured, a node name can appear in this display (as well as a node address). For instance, in the example display output below the node name `urk` is listed:

```
Neighbor 4172.224, TokenRing0, uptime 63:36:19, updated 8 secs ago
        The neighbors address is 4172.224, and named urk.
        We have sent queries for 0 nets via 0 packets.
        We received 0 replies and 0 extended replies.
        We have received queries for 143 nets via 12 packets.
        We sent 12 replies and 60 extended replies.
        We received 0 ZIP notifies.
        We received 0 obsolete ZIP commands.
        We received 4 miscellaneous ZIP commands.
We received 0 unrecognized ZIP commands.
We have received 44856 routing updates.
Of the 44856 valid updates, 0 entries were invalid.
We received 0 routing updates which were very late.
Last update had 0 extended and 1 non-extended routes.
Last update detail: 1 old
The neighbor has restarted 2 times in 125:17:24.
Cached service names for urk:
   urk:ciscoRouter@Low End SW Lab, socket 254
```

Note: The cached service names, which are used to determine the router name, and the neighbor's name (listed with its address), are only listed when **appletalk lookup-type** is enabled.

Displaying the Network Routing Table

To show the routing table for networks, use the **show appletalk route** EXEC commands:

show appletalk route [*network*]
show appletalk route [*interface-name*]

This command displays either the full routing table or just the entry for the optionally specified *network* for both extended and nonextended AppleTalk networks. For the extended AppleTalk networks, the command also displays cable ranges information.

The optional *interface-name* argument specifies an interface name to report on. Displays for both nonextended and extended AppleTalk networks follow.

When an AppleTalk route is poisoned by another router, its metric gets changed to poisoned (that is, 31 hops). The router will then age this route normally, during a hold-down period, when it will still be visible in the routing table with a distance of poisoned, or 31 hops.

A sample display for a nonextended AppleTalk network:

```
Codes: R - RTMP derived, C - connected, S - static, 3 routes
C Net 258 directly connected, 1431 uses, Ethernet0, zone Twilight
R Net 6 [1/G] via 258.179, 8 sec, 0 uses, Ethernet0, zone The O
C Net 11 directly connected, 472 uses, Ethernet1, zone No Parking
R Net 2154 [1/G] via 258.179, 8 sec, 6892 uses, Ethernet0, zone Local-
Talk
S Net 1111 via 258.144, 0 uses, Ethernet0, no zone set

[hops/state] state can be one of G:Good, S:Suspect, B:Bad
```

In the above display, the G rating after Net 6 indicates *good*. Alternate ratings are S for *suspect* and B for *bad*. These ratings are attained from the routing updates which occur at ten-second intervals. A separate and nonsynchronized event occurs at 20-second intervals, checking and flushing the ratings for particular routes that have not been updated. For each 20-second period that passes with no new routing information, a rating will slip from G to S to B; after one minute with no updates, that route will be flushed. Every time the router receives a useful update, the status of the route in question is reset to G. Useful updates are those advertising a route that is as good or better than the one currently in the table.

Following is a sample display for the extended AppleTalk network. Note the cable range display for Magnolia Estates:

```
Codes: R - RTMP derived, C - connected, 29 routes in internet

R Net 3 [1/G] via 254.163, 8 sec, Ethernet1, zone Localtalk
C Net 4 directly connected, Ethernet0, zone Twilight
C Net 6 directly connected, Ethernet3, zone Heavenly
R Net 11 [3/G] via 254.163, 8 sec, Ethernet1, zone UDP
R Net 17 [1/G] via 254.163, 8 sec, Ethernet1, zone UDP
R Net 33 [1/G] via 4.129, 1 sec, Ethernet0, zone Twilight
R Net 36 [1/G] via 254.174, 7 sec, Ethernet1, zone idontcare
R Net 55 [1/G] via 254.130, 9 sec, Ethernet1, zone Hospital
R Net 69 [1/G] via 4.129, 1 sec, Ethernet0, zone Empty Guf
R Net 70 [1/G] via 254.247, 2 sec, Ethernet1, zone Empty Guf
C Net 80 directly connected, Ethernet4, zone Light
R Net 99 [2/G] via 4.129, 1 sec, Ethernet0, zone BammBamm
C Net 254 directly connected, Ethernet1, zone Twilight
R Net 890 [2/G] via 4.129, 1 sec, Ethernet0, zone release lab
R Net 901 [2/G] via 4.129, 1 sec, Ethernet0, zone Dave's House
C Net 999-999 directly connected, Serial3, zone Magnolia Estates
R Net 2003 [4/G] via 80.129, 6 sec, Ethernet4, zone Bldg-13
R Net 2004 [2/G] via 80.129, 6 sec, Ethernet4, zone Bldg-17
R Net 2012 [2/G] via 4.130, 7 sec, Ethernet0, zone Bldg-13
R Net 2013 [3/G] via 254.163, 8 sec, Ethernet1, zone UDP
R Net 2024 [4/G] via 80.129, 3 sec, Ethernet4, zone Bldg-17
R Net 3004 [1/G] via 80.129, 3 sec, Ethernet4, zone Bldg-17
R Net 3012 [1/G] via 4.130, 5 sec, Ethernet0, zone Bldg-13
R Net 3024 [4/G] via 80.129, 3 sec, Ethernet4, zone Bldg-17
R Net 3880 [1/G] via 999.2, 0 sec, Serial3, zone Magnolia Estates
R Net 5002 [2/G] via 80.129, 3 sec, Ethernet4, zone Bldg-17
R Net 5003 [2/G] via 4.130, 5 sec, Ethernet0, zone Bldg-13
R Net 5006 [4/G] via 80.129, 3 sec, Ethernet4, zone Bldg-17
R Net 51489 [3/G] via 4.129, 8 sec, Ethernet0, zone Dave's House
```

Depending on the configuration of the global configuration commands **appletalk lookup-type** and **appletalk name-lookup-interval**, a node name can appear in this display (instead of a node address). For instance, in the example display output below, the node name gatekeeper is listed:

```
Codes: R - RTMP derived, C - connected, 67 routes in internet

R Net 3 [2/G] via gatekeeper, 4 sec, Ethernet0, zone Engineering
R Net 4 [3/G] via gatekeeper, 4 sec, Ethernet0, zone Twilight
R Net 6 [4/G] via gatekeeper, 4 sec, Ethernet0, zone Heavenly
R Net 11 [4/G] via gatekeeper, 4 sec, Ethernet0, zone UDP
R Net 12-12 [3/G] via gatekeeper, 4 sec, Ethernet0, zone UDP
```

```
R Net 17-17 [2/G] via gatekeeper, 4 sec, Ethernet0, zone Twilight
R Net 19-19 [3/G] via gatekeeper, 4 sec, Ethernet0, zone customer eng
R Net 29-29 [1/G] via gatekeeper, 4 sec, Ethernet0, zone Engineering
R Net 33 [2/G] via gatekeeper, 4 sec, Ethernet0, zone Twilight
R Net 69-69 [2/G] via gatekeeper, 4 sec, Ethernet0, zone Empty Guf
R Net 80 [3/G] via gatekeeper, 4 sec, Ethernet0, zone Light
R Net 199-199 [6/G] via gatekeeper, 4 sec, Ethernet0, zone Tir'n na'Og
R Net 550 [4/G] via gatekeeper, 4 sec, Ethernet0, zone outside cisco
R Net 560 [4/G] via gatekeeper, 4 sec, Ethernet0, zone outside cisco
R Net 666-666 [2/G] via gatekeeper, 4 sec, Ethernet0, zone Gates of Hell
R Net 2010 [7/G] via gatekeeper, 4 sec, Ethernet0, zone europe
R Net 2500-2500 [6/G] via gatekeeper, 4 sec, Ethernet0, zone Looking
Glass
C Net 2501-2501 directly connected, Ethernet1, no zone set
R Net 3004 [3/G] via gatekeeper, 4 sec, Ethernet0, zone Bldg-17
R Net 3010 [6/G] via gatekeeper, 4 sec, Ethernet0, zone europe
```

The next sample shows the result of the **show appletalk route** command with a specific network.

For the command **show appletalk route 69**, the display looks like this:

```
Codes: R - RTMP derived, C - connected, 67 routes in internet

R Net 69-69 [2/G] via gatekeeper, 0 sec, Ethernet0, zone Empty Guf
  Route installed 125:20:21, updated 0 secs ago
  Next hop: gatekeeper, 2 hops away
  Zone list provided by gatekeeper
  Route has been updated since last RTMP was sent
  Valid zones: "Empty Guf"
```

Depending on the configuration of the global configuration commands **appletalk lookup-type** and **appletalk name-lookup-interval**, a node name can appear in this display (instead of a node address). For instance, in the example display output above, the node name gatekeeper is listed.

For the command **show appletalk route serial 3**, the display looks like this:

```
Codes: R - RTMP derived, C - connected, 29 routes in internet

C Net 999 directly connected, Serial3, zone Magnolia Estates
R Net 3880 [1/G] via 999.2, 3 sec, Serial3, zone Magnolia Estates
```

Displaying Information About the Sockets

The command **show appletalk socket** displays information about the process-level processing in all the sockets in the AppleTalk interface. Enter this command at the EXEC prompt:

show appletalk socket [*socket-number*]

When used with the optional *socket-number* argument, it shows information about a specific socket.

The following is the output seen when no socket number is specified:

```
Socket  Name            Owner                   Waiting/Processed

    1   RTMP             AT RTMP                  0 148766
    2   NIS              AT NBP                   0 156429
    4   AEP              AT Maintenance           0      0
    6   ZIP              AT ZIP                   0  13619
    8   SNMP             AT SNMP                  0      0
  253   PingServ         AT Maintenance           0      0
```

When a socket is specified, only statistics for that socket are displayed, as seen in following sample output:

```
 6   ZIP              AT ZIP                   0   2704
```

Displaying AppleTalk Traffic Information

The EXEC command **show appletalk traffic** displays AppleTalk-specific traffic information. The command has this syntax:

show appletalk traffic

The statistics it displays include the total number of packets received, categorized errors, summaries of packets received for the various AppleTalk services (for example, NBP, ZIP, DDP) and for other protocols such as Echo and ARP. Several counters have also been added to monitor extended AppleTalk activity. See Table 1-13.

Following is a sample display of extended AppleTalk activity.

```
AppleTalk statistics:
  Rcvd:  357471 total, 0 checksum errors, 264 bad hop count
         321006 local destination, 0 access denied
         0 for MacIP, 0 bad MacIP, 0 no client
         13510 port disabled, 2437 no listener
         0 ignored, 0 martians
  Bcast: 191881 received, 270406 sent
  Sent:  550293 generated, 66495 forwarded, 1840 fast forwarded
         0 forwarded from MacIP, 0 MacIP failures
         436 encapsulation failed, 0 no route, 0 no source
  DDP:   387265 long, 0 short, 0 macip, 0 bad size
  NBP:   302779 received, 0 invalid, 0 proxies
        57875 replies sent, 59947 forwards, 418674 lookups, 432 failures
  RTMP:  108454 received, 0 requests, 0 invalid, 40189 ignored
         90170 sent, 0 replies
  ATP:   0 received
  ZIP:   13619 received, 33633 sent, 32 netinfo
  Echo:  0 received, 0 discarded, 0 illegal
         0 generated, 0 replies sent
  Responder:  0 received, 0 illegal, 0 unknown
         0 replies sent, 0 failures
  AARP:  85 requests, 149 replies, 100 probes
         84 martians, 0 bad encapsulation, 0 unknown
         278 sent, 0 failures, 29 delays, 315 drops
  Lost: 0 no buffers
  Unknown: 0 packets
  Discarded: 130475 wrong encapsulation, 0 bad SNAP discriminator
```

Table 1-13 Show Apple Traffic Field Descriptions

Field	Description
checksum errors	The DDP checksum was incorrect so these packets were discarded. The DDP checksum is verified for packets which are directed to the router. Forwarded packets do not have their checksums verified enroute.
bad hop count	Packet dropped, the packet has travelled too many hops.
local destination	The number of packets that were received for processing by the router.
access denied	Packet dropped, access list did not permit it.
no client	The number of packets that were directed to a MacIP client but that was not present. The packets were discarded.
port disabled	Packet dropped, routing disabled for port (extended AppleTalk only). Occurs because of a configuration error or a packet received while in verification/discovery mode.
no listener	The number of packets directed to a socket on the router that does not have any services associated with that socket. The packets were discarded.
ignored	The number of routing update packets that were ignored because the packet was from a misconfigured neighbor. Also, packets are ignored when routing is disabled.
martians	The number of packets that were discarded because they contained bogus information in the DDP header. What distinguishes this error from the others is that the data in the header is never valid as opposed to not being valid at a given point in time.
fast forwarded	Packets which were forwarded using data from the fast switching (route cache). These packets incur the least delay and cause the least impact with respect to the router.
encapsulation failed	Packet received for a connected network, but node's MAC address not found.
bad size	Physical packet length and claimed length disagree.
netinfo	Number of packets which requested port configuration via ZIP GetNetInfo requests. Originally, these were exclusively used during node startup, but are now used be some AppleTalk network management software packages.
unknown	Unknown AppleTalk packet type.
no buffers	Attempted packet buffer allocation failed.

Field	Description
wrong encapsulation	Nonextended AppleTalk packet on extended AppleTalk port, or vice versa.
bad SNAP discriminator	Extended AppleTalk packet without Apple discriminator (extended AppleTalk only). Occurs when another AppleTalk device has implemented an obsolete or incorrect packet format.

Displaying Zone Information

The **show appletalk zone** command displays the zone information table and has this syntax:

show appletalk zone [*zonename*]

Use this command to display which networks comprise each zone for both nonextended and extended AppleTalk networks.

The argument *zonename* specifies the name of the zone you are trying display information on.

In the following sample display, notice the report of cable ranges for the extended zone Empty Guf:

```
Name                                Network(s)
Gates of Hell                       666-666
Engineering                         3 29-29 4042-4042
customer eng                        19-19
CISCO IP                            4140-4140
Dave's House                        3876 3924 5007
Narrow Beam                         4013-4013 4023-4023 4037-4037 4038-4038
Low End SW Lab                      6160 4172-4172 9555-9555 4160-4160
Tir'n na'Og                         199-199
Mt. View 1                          7010-7010 7122 7142 7020-7020 7040-7040
                                    7060-7060
Mt. View 2                          7152 7050-7050
UDP                                 11 12-12
Empty Guf                           69-69
Light                               80
europe                              2010 3010 3034 5004
Bldg-13                             4032 5026 61669 3012 3025 3032 5025 5027
Bldg-17                             3004 3024 5002 5006
S/W Test Lab                        27001-27001
Dead Ringer                         4028-4028 4035-4035 4036-4036
outside cisco                       550 560 4014-4014 4020-4020
Pin Point                           25346 25344 25345-25345
```

If a specific *zonename* is specified, the display output appears as follows:

```
AppleTalk Zone Information for CISCO IP:
  Valid for nets: 4140-4140
  Not associated with any interface.
  Not associated with any access list.
```

Maintaining the AppleTalk Network

Cisco provides two EXEC commands to clear the different AppleTalk data structures.

Clearing the Neighbor Data Structures

The **clear appletalk neighbors** command clears the AppleTalk neighbors data tructures. Enter this command at the EXEC prompt:

clear appletalk neighbors

Clearing the Router Data Structures

The **clear appletalk routes** command clears the AppleTalk route data structures. Enter this command at the EXEC prompt:

clear appletalk routes

The AppleTalk Ping Command

The EXEC **ping** command sends Echo Protocol datagrams to other AppleTalk nodes to verify connectivity and measure round-trip times.

When the **ping** command prompts for a protocol, specify **appletalk**. Default options are indicated with carriage returns. What follows is a sample of using **ping** with the AppleTalk protocol. To abort a ping session, type the escape sequence (by default, type Ctrl-^X, which is done by simultaneously pressing the Ctrl, Shift, and 6 keys, letting go then pressing the X key).

Sample Session:

```
Protocol [ip]: appletalk
Target Appletalk address: 1024.128
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Verbose [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte AppleTalk Echos to 1024.128, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 4/4/8 ms
```

Note: Only an interface that supports *HearSelf* can respond to packets generated at a local console and directed to an interface on the same router. Cisco routers only support *HearSelf* on Ethernet.

The **ping** command uses the characters in Table 1-14 to indicate the success or failure of each packet in the **ping** sequence.

Table 1-14 AppleTalk Ping Characters

Character	Meaning
!	The packet was echoed successfully from the target address.
.	The timeout period expired before an echo was received from the target address.
B	Bad, or malformed echo was received from the target address.
C	An echo was received with a bad DDP checksum.
E	Transmission of the echo packet to the target address failed.
R	The transmission of the echo packet to the target address failed for lack of a route to the target address.

AppleTalk NBP Ping Interface

The **ping** EXEC command for AppleTalk allows testing and informational lookup of NBP registered entities.

To use this privileged facility, type **ping** and respond to the protocol prompt with the keyword **appletalk**. Then enter the keyword **nbp** in response to the prompt `Target AppleTalk address:`

The following is an example of the sequence used to initialize the AppleTalk *nbptest* utility accessed via the **ping** command.

```
myrouter# ping
Protocol [ip]: appletalk
Target AppleTalk address: nbp
nbptest>
```

The *nbptest* facility is an interactive, menu-driven facility. Type `help` or `?` to see the command list. Type `quit` to return to the EXEC prompt. The sections that follow describe the subcommands available from the *nbptest* utility invoked with this command.

Help Subcommand

The **help** subcommand of the *nbptest* utility displays the available tests in a menu.

The following is a sample of the menu displayed:

```
nbptest> help
Tests are:

lookup:     lookup an NVE.  prompt for name, type and zone
parms:      display/change lookup parms (ntimes, nsecs, interval)
zones:      display zones
poll:       for every zone, lookup all devices, using default parms
help|?:     print command list
confirm:    confirm an NVE.  prompt for name, type, zone, address
addclients: add a range of fake MACIP clients
delclients: delete a range of fake MACIP clients
register:   register an NVE.  prompt for name, type, zone
unregister: unregister an NVE.  prompt for name, type, zone
stats:      dump appletalk stats changed since last dump
debug:      set/unset debug switches
quit:       exit nbptest
```

Parms Subcommand

The **parms** subcommand of the *nbptest* utility sets the *lookup* parameters used in subsequent lookup and poll commands.

The following is an example parameter configuration sequence.

```
nbptest> parms
maxrequests [10]: 1
maxreplies [5]: 100
interval [5]: 10
```

Note: If the values of the **parms** subcommand are revised, the next time this menu is activated, the parms last entered appear is brackets.

In the above example, the number of lookup retries is set to 1, the maximum number of replies to accept for each lookup is set to 100, and the interval between each retry is set to 10 seconds.

The defaults for `maxrequests`, `maxreplies`, and `interval` are 10, 5, and 5, respectively; the current value is indicated in brackets in the prompt for each parameter.

The acceptable ranges are as follows:

- `maxrequests`—1 to 5 (integer value) requests
- `maxreplies`—1 to 500 (integer value) replies
- `interval`—1 to 60 (integer value) seconds

Lookup Subcommand

Use the **lookup** subcommand to search for NBP entities in a specific zone. The **parms** command can be used to adjust the lookup parameters. Nonprinting characters can be specified by entering a three-character string specifying the hexadecimal equivalent (for example, `:c5` specifies the NBP truncation wildcard).

Example:

The following example sequence illustrates the specification of the **parm** subcommand parameter.

```
nbptest> parms
maxrequests [10]: 1
maxreplies [5]: 100
interval [5]: 10

nbptest> lookup
Entity name [=]:
Type of Service [ipgateway]:  macintosh:c5
Zone [bldg-17]:  engineering
(100n,50a,253s)[1]:   'userA:Macintosh IIcx@engineering'
(100n,16a,251s)[1]:   'userB:Macintosh II@engineering'
(200n,24a,253s)[1]:   'userC:Macintosh IIci@engineering'
(200n,36a,253s)[1]:   'userD:Macintosh IIci@engineering'
(300n,21a,252s)[1]:   'userE:Macintosh SE/30@engineering'
(300n,97a,251s)[1]:   'userF:Macintosh SE/30@engineering'
NBP lookup request timed out
Processed 6 replies, 7 events
```

The AppleTalk DDP address of the registered entity is displayed in parenthesis, (network, node address, and socket number), followed by the NBP enumerator and the NBP entity string.

Note: If the values of the **parms** subcommand are revised, the next time this menu is activated, the parameters last entered appear is brackets.

Poll Subcommand

Use the **poll** command to search for all devices in all zones according to the current lookup parameters. The poll command posts a lookup of the form “=:=@zone” for each zone in the AppleTalk internet.

In a large AppleTalk internetwork, the **poll** subcommand will return several hundred replies and generate a large amount of network activity, and so should be used with caution.

The following is a sample output for this command:

```
poll:  sent 2 lookups
(100n,82a,252s)[1]:   'userA:Macintosh IIci@Zone one'
(200n,75a,254s)[1]:   'userB:Macintosh IIcx@Zone two'
```

```
NBP polling completed.
Processed 2 replies, 2 events
```

The AppleTalk DDP address of the registered entity is displayed in parentheses, (network, node address, and socket number), followed by the NBP enumerator and the NBP entity string.

Zones Subcommand

The **zones** subcommand displays the current zone list in the router. It is equivalent to the **show appletalk zones** EXEC command, and is included in *nbptest* for convenience.

The following is a sample output for this command:

```
Name                                     Network(s)
UDP                                      17 11
Heavenly                                 1161 6
Hospital                                 55
Bldg-17                                  82 81 14 13
CSL EtherTalk                            22
Twilight                                 1544 254 36 33 4
EtherTalk                                2
Underworld                               666
Magnolia Estates                         3880 999
Light                                    80
LocalTalk                                3
Empty Guf                                69-69
Total of 12 zones
```

Debugging the AppleTalk Network

The EXEC **debug** commands described in this section are used to troubleshoot the AppleTalk network transactions. Generally, you enter these commands during troubleshooting sessions with Cisco customer engineers.

For each **debug** command, there is a corresponding **undebug** command that turns the display off. Remember that some of these commands can be entered in groups that then display additional information.

debug apple-arp

The **debug apple-arp** command enables debugging of AppleTalk address resolution protocol. A side effect of enabling this option is that gleaning MAC information from datagrams is disabled.

debug apple-errors

The **debug apple-errors** command reports information about errors that occur. The information displayed by this command is enhanced by enabling debugging for the specific class of errors that you are interested in. This is similar to **debug apple-packets**.

debug apple-event

The **debug apple-event** command displays debugging information about AppleTalk special events, neighbors becoming reachable/unreachable, and interfaces going up/down. Only significant events (for example, neighbor and/or route changes) are logged. This command is maintained in nonvolatile memory, if present.

appletalk event-logging

The **appletalk event-logging** configuration command causes logging of a subset of messages produced by **debug appletalk** command. Logs significant events using the logger facility. Logged events include routing changes, zone creation, port status, and address.

debug apple-nbp

The **debug apple-nbp** command enables debugging output from the Name Binding Protocol (NBP) routines.

debug apple-packet

The **debug apple-packet** command enables per-packet debugging output. It reports information online when a packet is received or a transmit is attempted. The command allows watching the types of packets being slow switched. It is roughly equivalent to turning on all the other AppleTalk debugging information. There will be at least one line of debugging output per AppleTalk packet processed.

The **debug apple-packet** command, when invoked in conjunction with the commands **debug apple-routing**, **debug apple-zip**, and **debug apple-nbp**, adds protocol processing information in addition to generic packet details. It reports protocol processing, and successful completion or failure information.

The **debug apple-packet** command, when invoked in conjunction with the command **debug apple-errors**, reports packet level problems such as encapsulation problems. This is the case because **debug apple-errors** is a subset of **debug apple-packets**.

debug apple-routing

The **debug apple-routing** command enables debugging output from the Routing Table Maintenance Protocol (RTMP) routines. This command can be used to monitor acquisition of routes, aging of routing table entries, and advertisement of known routes. It also reports conflicting network numbers on the same network if the network is misconfigured.

debug apple-zip

The **debug apple-zip** command enables debugging output from the Zone Information Protocol routines. This command reports significant events such as discovery of new zones and zone list queries.

AppleTalk Global Configuration Command Summary

This section lists all the global commands used with the AppleTalk interface.

[**no**] **access-list** *list* {**permit**|**deny**} **network** *network*
[**no**] **access-list** *list* {**permit**|**deny**} **cable-range** *start-end*
[**no**] **access-list** *list* {**permit**|**deny**} **includes** *start-end*
[**no**] **access-list** *list* {**permit**|**deny**} **within** *start-end*
[**no**] **access-list** *list* {**permit**|**deny**} **zone** *zonename*
no access-list *list*
access-list *list* {**permit**|**deny**} **additional-zones**
access-list *list* {**permit**|**deny**} **other-access**

Defines an AppleTalk access list. This command has several optional formats and supports *extended* AppleTalk networks. The argument *list* is an integer from 600 to 699 and the argument *network* is an AppleTalk network. number. Additional **permit** and **deny** conditions may be added to the list by issuing further **access-list** commands for that list. Use the **no access-list** command with the *list* number only to remove an entire access list from the configuration. Specify the optional arguments to remove a particular clause.

no appletalk arp

Resets the **arp interval** and **arp retransmit** commands to their default values.

appletalk arp {**request**|**probe**} **interval** *milliseconds*

Specifies the time interval between retransmission of ARP packets. The argument *milliseconds* specifies the interval. The default is 200 when the **probe** keyword is used and 1000 when the **request** keyword is used. The minimum value is 33 milliseconds. The command **no appletalk arp** or a *milliseconds* value of 0 resets the default.

appletalk arp {**request**|**probe**} **retransmit-count** *count*

Specifies the number of retransmissions that will be done before abandoning address negotiations and using the selected address. The argument *count* specifies the retransmission count. The default is 10 when the **probe** keyword is used and 5 when the **request** keyword is used.The minimum value that can be specified is 1 (one). The command **no appletalk arp** or a *count* value of 0 resets the default.

[**no**] **appletalk checksum**

Enables and disables the generation and verification of checksums for all AppleTalk packets (except routed packets) when enabled. An incoming packet with a nonzero checksum will be verified against that checksum and discarded if in error. By default, checksum verification is enabled.

[**no**] **appletalk event-logging**

Causes logging of a subset of messages produced by **debug appletalk** command. The **no** form of the command turns this function off. Logs significant events using the logger facility. Logged events include routing changes, zone creation, port status, and address.

appletalk iptalk-baseport *port-number*

Specifies the UDP port number, which is the beginning of the range of UDP ports used in mapping AppleTalk well-known DDP socket numbers to UDP ports. The argument *port-number* is the first UDP port number.

[**no**] **appletalk lookup-type** *serviceType*

Specifies services listed in **show appletalk nbp** and **show appletalk name-cache** EXEC command display. The argument *serviceType* is the specific AppleTalk service. The command **no appletalk lookup-type** can be used with or without the *serviceType* argument. Using the argument specifies exclusion of a specific service type from the name cache. Prevent all names (except those relating to Cisco routers) from being cached by using the **no** version of this command without the argument *serviceType*.

[**no**] **appletalk macip dynamic** *ip-address* [*ip-address*] **zone** *server-zone*

Allocates a single IP address or a range of IP addresses to be assigned to *dynamic* MacIP clients by the MacIP server serving zone *server-zone*. Dynamic clients are those who accept *any* IP address assignment within the dynamic range specified. The **no appletalk macip** command shuts down all running MacIP services. If entered with the keyword **dynamic**, a specific *ip-address* range and a specific *server-zone*, the particular dynamic address assignment statement (if one exists) will be eliminated from the configuration.

[**no**] **appletalk macip server** *ip-address* **zone** *server-zone*

Establishes a new MacIP server. Only one MacIP server can be configured per AppleTalk zone. A server is not registered via NBP until at least one MacIP resource is configured. The **no appletalk macip** command shuts down all active MacIP services. If entered with the keyword **server**, a specific *ip-address* and a specific *server-zone*, the particular server statement (if one exists) will be shutdown and eliminated from the configuration.

[**no**] **appletalk macip static** *ip-address* [*ip-address*] **zone** *server-zone*

Defines a range of addresses to be made available to MacIP clients who have reserved an invariant IP address. The server keeps track of these address for routing and informational purposes. The **no appletalk macip** command shuts down all running MacIP services. If entered with the keyword **static**, a specific *ip-address* and a specific *server-zone*, the particular static address assignment statement (if one exists) will be eliminated from the configuration.

[**no**] **appletalk name-lookup-interval** *intInSeconds*

Sets the interval between service pollings by the router on its AppleTalk interfaces. The argument *intInSeconds* is the interval in seconds between NBP lookup pollings. A value of zero (0) is equivalent to **no appletalk name-lookup-interval**. Both disable name lookup. The default is zero (0). You cannot disable lookup of **ciscoRouter**.

[**no**] **appletalk permit-partial-zones**

Allows access to zones that contain networks that do not have direct access. In other words, when a specific zone is *partially* obscured, other (visible) networks that are not subject to access control are propagated normally when **permit-partial-zones** is enabled. The default is for **appletalk permit-partial-zones** to be *disabled*.The **no appletalk permit-partial-zones** version of this command disables this option, and restores the default condition where a complete zone is controlled if any associated network is controlled. If this command is enabled, networks for the zone are propagated, even if one or more networks are access-controlled.

[**no**] **appletalk proxy-nbp** *network-number zonename*

Required for each zone that has a nonextended-only AppleTalk router connected to a network in the zone. The argument *network-number* must be a unique network number which will be advertised via this router as if it were a real network. The argument *zonename* is the name of the zone requiring compatibility support. Only one proxy is needed to support a zone, but additional proxies can be defined with different network numbers if redundancy is desired The **no** version removes the specified network/zone association.

[**no**] **appletalk require-route-zones**

Prevents *bogus* routes (possibly generated by a broken router or corrupt packet) from causing ZIP protocol storms. The default is for **require-route-zones** to be *enabled*. When **require-route-zones** is enabled, the router will not advertise a route to its neighbors until it has obtained the network/zone associations. Use the **no appletalk require-route-zones** command to disable the **requ ire-route zones** option and set the condition such that the router can advertise routes to its neighbors without having obtained the network-zone associations.

[**no**] **appletalk routing**

Enables or disables the AppleTalk protocol processing.

[**no**] **appletalk strict-rtmp**

Enforces maximum checking of routing packets to ensure their validity. The default of this command is to provides maximum checking. The **no** variation disables the maximum checking mode.

[**no**] **appletalk timers** *update-interval valid-interval invalid-interval*

Changes the time intervals (in seconds) used in AppleTalk routing. The argument *update-interval* is the time between routing updates sent to other routers on the network; the default is 10 seconds. The argument *valid-interval* is amount of time that the router will consider a route valid without having heard a routing update for that route; the default is 20 seconds, and the value is normally twice the update interval. The argument *invalid-interval* is the amount of time that the router will wait before marking a route invalid; the default is three times the *valid-interval*, or 60 seconds.

AppleTalk Interface Subcommand Summary

This section lists, in alphabetical order, all the interface subcommands used with AppleTalk networks.

[**no**] **appletalk access-group** *list*

Assigns an interface to an access list. The argument *list* specifies the appropriate AppleTalk access list. Use the **no** form of the command to remove the list from the interface.

[**no**] **appletalk address** *address*

Assigns AppleTalk addresses on the interfaces that will be used for the AppleTalk protocol. Use this command prior to assigning zone names. Use this subcommand to configure nonextended interfaces. The **no** version removes the specified address.

[**no**] **appletalk cable-range** *start-end* [*network.node*]

Designates an interface as being on an extended AppleTalk network. This range is specified using th*e start-end* parameter, which is a pair of decimal numbers between 1 and 65279, inclusive. The starting and ending addresses should usually be assigned equal numbers. The optional *network.node* argument specifies the suggested network and node number that will be used first when selecting the AppleTalk address for this interface. The **no** version removes the specified cable range.

[**no**] **appletalk discovery**

Resets the discovery mode and allows a new cable range to be discovered. Use the **no** variation to return the software to the default (off) state.

[**no**] **appletalk distribute-list** *access-list-number* **in**

Filters input from networks. The argument *access-list-number* is the number of a predefined access list. The keyword **in** is used to filter networks received in update. The **no** version removes the specified distribution list.

[**no**] **appletalk distribute-list** *access-list-number* **out**

Filters output from networks. The argument *access-list-number* is the number of a predefined access list. The keyword out is used to suppress networks from being sent in updates. The **no** version removes the specified distribution list.

[**no**] **appletalk getzonelist-filter** *access-list-number*

Modifies zone-list replies. The argument *access-list-number* must be in the range of 600 to 699, inclusive. If an undefined access list is used, the rule defaults to **permit**. If a zone does not match any rule in the list, it is denied, unless permitted via the **additional-zones** option of the **access-list** global configuration command. Use the **no appletalk getzonelist** *access-list-number* command to remove this filter. Numeric entries in the access list are ignored by this filter.

appletalk iptalk *net.node zone*

Encapsulates AppleTalk in IP packets in a manner compatible with the Columbia AppleTalk Package (CAP) IPTalk and the Kinetics IPTalk (KIP) implementations. This command enables IPTalk encapsulation on an interface that already has an configured IP address. The argument *net.node* is a network node address; the argument *zone* is the AppleTalk zone.

[**no**] **appletalk send-rtmp**

Allows a router to be placed on a net with AppleTalk enabled, but without being seen. This allows disabling of routing update. The default is to send updates. The **no** version blocks updates from being sent.

[**no**] **appletalk zone** *zonename*

Sets the zone name for the connected AppleTalk network. This command also specifies the zone name associated with the AppleTalk network for the specified interface. The argument *zonename* specifies the name of the zone for the connected AppleTalk network. The argument is ignored for nonextended AppleTalk. The command is ignored if the specified zone name is not in the zone list. The **no** form of the command deletes a zone name from a zone list or the entire zone list if none is specified. Must be specified after the **appletalk address** or **appletalk cable-range** command if discovery is not enabled. This command may be issued multiple times if it follows the **appletalk cable-range** command.