CHAPTER 17

Configuring AppleTalk Routing

This chapter has been added since publication of the IOS 10 Release documents. AppleTalk routing is now supported on communication servers.

Note This chapter discusses routing functionality of your communication server. For simplicity, we have used the term *communication server* to indicate those servers which provide routing, and therefore act as routers.

AppleTalk is a local-area network (LAN) system that was designed and developed by Apple Computer, Inc. It can run over Ethernet, Token Ring, and Fiber Data Distributed Interface (FDDI) networks and over Apple's proprietary twisted-pair media access system (LocalTalk). AppleTalk specifies a protocol stack comprising several protocols that direct the flow of traffic over the network.

Apple Computer uses the name *AppleTalk* to refer to the Apple network protocol architecture. Apple Computer refers to the actual transmission media used in an AppleTalk network as LocalTalk, TokenTalk (AppleTalk over Token Ring), EtherTalk (AppleTalk over Ethernet), and FDDITalk (AppleTalk over FDDI).

This chapter describes how to configure AppleTalk and provides configuration examples. For a complete description of the commands mentioned in this chapter, refer to the "AppleTalk Routing Commands" chapter in the *Access and Communication Servers Command Reference* publication. For historical background and a technical overview of AppleTalk, see the *Internetworking Technology Overview* publication.

Cisco's Implementation of AppleTalk

Cisco communication servers and IOS software support AppleTalk Phase 1 and AppleTalk Phase 2. For AppleTalk Phase 2, Cisco IOS software supports both *extended* and *nonextended* networks. Cisco's implementation of AppleTalk can route packets over Ethernet, Token Ring, and FDDI LANs, and over X.25, High-Level Data Link Control (HDLC), Frame Relay, and Switched Multimegabit Data Service (SMDS) wide-area networks (WANs).

Cisco IOS software also supports AppleTalk Enhanced Internet Gateway Routing Protocol (IGRP). AppleTalk Enhanced IGRP provides the following features:

- Automatic redistribution. By default, AppleTalk Routing Table Maintenance Protocol (RTMP) routes are automatically redistributed into Enhanced IGRP, and AppleTalk Enhanced IGRP routes are automatically redistributed into RTMP. If desired, you can turn off redistribution. You also can completely turn off AppleTalk Enhanced IGRP and AppleTalk RTMP on the communication server or on individual interfaces.

- Configuration of routing protocols on individual interfaces.You can configure interfaces that have been configured for AppleTalk to use either RTMP, Enhanced IGRP, or both routing protocols. If two neighboring communication servers are configured to used both RTMP and Enhanced IGRP, the Enhanced IGRP routing information will supersede the RTMP information. However, both communication servers will continue to send RTMP routing updates. This feature allows you to control the excessive bandwidth usage of RTMP on WAN links. Because a WAN link is a point-to-point link, there are no other devices on the link, and hence, there is no need to run RTMP to perform end-node discovery. Using Enhanced IGRP on WAN links allows you to save bandwidth and, in the case of PSDNs, traffic charges.

Standard AppleTalk Services

The Cisco implementation of AppleTalk supports the following standard AppleTalk protocols:

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

AppleTalk ARP, DDP, and RTMP provide end-to-end connectivity between internetworked nodes. AppleTalk ARP maps AppleTalk node addresses to the addresses of the underlying data link, thus making it possible for AppleTalk to run on several data links. DDP provides socket-to-socket delivery of packets. RTMP establishes and maintains routing tables.

NBP and ZIP maintain node name and zone information. NBP maps network names to AppleTalk addresses. ZIP tracks which networks are in which zones.

AEP is an echo, or **ping**-type, protocol. It generates packets that test the reachability of network nodes.

ATP is a reliable transport protocol that provides data acknowledgment and retransmission for transaction-based applications, such as file services provided by the AppleTalk Filing Protocol (AFP) and print services provided by the Printer Access Protocol (PAP).

Our software provides support for the AppleTalk MIB variables as described in RFC 1243. We provide support for the following AppleTalk protocols: AppleTalk ARP, AppleTalk Port Group, AppleTalk Datagram Delivery Protocol (DDP), AppleTalk Routing Table Maintenance Protocol (RTMP), AppleTalk Zone Information Protocol (ZIP), AppleTalk Name Binding Protocol (NBP), and AppleTalk Echo Group.

Enhancements to Standard AppleTalk

The Cisco AppleTalk implementation includes the following enhancements to standard AppleTalk:

- Support for EtherTalk 1.2 and EtherTalk 2.0 without the need for translation or transition communication servers.
- Support for WAN protocols, including SMDS, Frame Relay, X.25, and HDLC.
- Configurable protocol constants (examples include controlling the aging of entries in the routing table and controlling the AARP interval and number of retransmissions).

- No software limits on the number of zones or routes.
- MacTCP support via a MacIP server.
- Support of IPTalk, which provides Internet Protocol (IP) encapsulation of AppleTalk, IPTalk, and the Columbia AppleTalk Package (CAP).
- Access control for filtering network traffic by network number, filtering routing table updates, and filtering GetZoneList (GZL) responses.
- Integrated node name support to simplify AppleTalk network management.
- Interactive access to AEP and NBP provided by the **ping** command.
- Configured (seed) and discovered interface configuration.
- Support for the AppleTalk Responder, which is used by network monitoring packages such as *Inter•Poll*.
- SNMP over AppleTalk.
- Encapsulation (tunneling) of AppleTalk RTMP packets over an IP backbone.
- Support for AppleTalk static routes.

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

Note that Cisco's implementation of AppleTalk does not forward packets with local source and destination network addresses. This behavior does not conform with the definition of AppleTalk in Apple Computer's *Inside AppleTalk* publication. However, this behavior is designed to prevent any possible corruption of the AppleTalk ARP table in any AppleTalk node that is performing MAC-address gleaning.

AppleTalk Phase 1 and Phase 2

There are two versions, or phases, of AppleTalk. AppleTalk Phase 1 and AppleTalk Phase 2 are implementations of the AppleTalk protocol stack, especially the routing portions of the stack.

AppleTalk Phase 1, the earlier version, supports a single physical network that can have one network number and be in one zone. This network can have up to 254 devices, which can consist of 127 end nodes and 127 servers. *AppleTalk Phase 2*, the more recent version, supports multiple logical networks on a single physical network. This means that one cable segment can have multiple network numbers. Each logical network in Phase 2 can support up to 253 devices, with no restrictions on the type of devices. Also, in AppleTalk Phase 2 a network can be in more than one zone.

AppleTalk Phase 2 introduced the concepts of *extended* and *nonextended* networks. These terms refer to the media-level encapsulation and cable addressing used on a network segment attached to a communication server interface. While the concepts of extended and nonextended networks do not exist in AppleTalk Phase 1, Phase 1 can be thought of as a nonextended network.

Table 17-1 compares the capabilities of AppleTalk Phase 1 and Phase 2.

Table 17-1 AppleTalk Phase 1 and Phase 2

Capability	AppleTalk Phase 1	AppleTalk Phase 2
Networks, nodes, and zones		
Number of logical networks (cable segments)	1	Unlimited
Maximum number of devices	254[1]	253[2]
Maximum number of end nodes	127	Does not apply[3]
Maximum number of servers	127	Does not apply
Number of zones in which a network can be	1[4]	1 (nonextended) 255 (extended)
Media-level encapsulation		
Nonextended network	Does not apply	Yes
Extended network	Does not apply	Yes
Cable addressing	Does not apply; uses network numbers	Single network number (nonextended) Cable range of 1 or more (extended)

1. The node addresses 0 and 255 are reserved.
2. The node addresses 0, 254, and 255 are reserved.
3. There is no restriction on the types of devices. There can be a total of 253 end nodes and servers.
4. In terms of zones, an AppleTalk Phase 1 network can be thought of as a nonextended AppleTalk Phase 2 network.

Communication Servers running Software Releases 8.2 and later support AppleTalk Phase 1 and Phase 2.

AppleTalk Addresses

An AppleTalk *address* consists of a network number and a node number expressed in decimal in the format *network.node*.

The *network number* identifies a network, or cable segment. A *network* is a single logical cable. Although the logical cable is frequently a single physical cable, bridges can be used to interconnect several physical cables. The network number is a 16-bit decimal number that must be unique throughout the entire AppleTalk internetwork. In AppleTalk Phase 1, networks are identified by a single network number that corresponds to a physical network. In AppleTalk Phase 2, networks are identified by a cable range that corresponds to one or more logical networks. In Phase 2, a single cable can have multiple network numbers. A cable range is either one network number or a contiguous sequence of several network numbers in the format *start–end*. For example, the cable range 4096–4096 identifies a logical network that has a single network number, and the cable range 10–12 identifies a logical network that spans three network numbers. In both AppleTalk Phase 1 and Phase 2, the network number 0 is reserved.

The *node number* identifies the node, which is any device connected to the AppleTalk network. The node number is an 8-bit decimal number that must be unique on that network. In AppleTalk Phase 1, node numbers 1 through 127 are for user nodes, node numbers 128 through 254 are for servers, and node numbers 0 and 255 are reserved. In AppleTalk Phase 2, you can use node numbers 1 through 253 for any nodes attached to the network. Node numbers 0, 254, and 255 are reserved.

The following is an example of an AppleTalk network address:

```
3.45
```

In this example, the network number is 3 and the node number is 45. You enter both numbers in decimal. Our software also displays them in decimal.

A *zone* is a logical group of networks. The networks in a zone can be contiguous or noncontiguous. A zone is identified by a zone name, which can be up to 32 characters long and can include standard characters as well as AppleTalk special characters. To include a special character, type a colon followed by two hexadecimal characters that represent the special character in the Macintosh character set. In AppleTalk Phase 2, an extended network can have up to 255 zones, and a nonextended network can have only one zone. An AppleTalk Phase 1 network can have only one zone.

Configuration Guidelines and Compatibility Rules

AppleTalk Phase 1 and AppleTalk Phase 2 networks are incompatible and cannot run simultaneously on the same internetwork. As a result, all communication servers and routers in an internetwork must support AppleTalk Phase 2 before the network can use Phase 2 routing. If your internetwork has a combination of AppleTalk Phase 1 and Phase 2 communication servers and routers, you must observe the compatibility rules described in this section. Note, however, that you do not need to upgrade all end nodes in order to use the features provided by our AppleTalk enhancements.

Follow these guidelines when configuring an extended AppleTalk network on our communication server if any router in your AppleTalk internetwork supports only nonextended AppleTalk (that is, if any routers are Phase 1 routers). If you do not follow these guidelines, unpredictable behavior might result.

- The cable range must be one (for example, 23–23).
- Each AppleTalk network can be a member of only one zone.

When using Cisco routers and communication servers with other vendors' implementations of AppleTalk, follow these guidelines:

- In order for a Macintosh with an Ethernet card to support extended AppleTalk, the Macintosh must be running EtherTalk Version 2.0 or later. This restriction does not apply to Macintoshes with only LocalTalk interfaces.
- Shiva FastPath routers must run K-Star Version 8.0 or later and must be explicitly configured for extended AppleTalk.
- Apple's Internet Router software Version 2.0 supports a transition mode for translation between nonextended AppleTalk and extended AppleTalk on the same network. Transition mode requires the Apple upgrade utility and a special patch file from Apple.

AppleTalk Configuration Task List

To configure AppleTalk routing, complete the tasks in the following sections. At a minimum, you must enable AppleTalk routing. The remaining tasks are optional.

- Enable AppleTalk Routing
- Create an AppleTalk Routing Process
- Control Access to AppleTalk Networks
- Configure the Name Display Facility
- Set up Special Configurations
- Configure AppleTalk Control Protocol for PPP

- Tune AppleTalk Network Performance
- Configure AppleTalk Enhanced IGRP
- Configure AppleTalk Inter-Enterprise Routing
- Configure AppleTalk over WANs
- Monitor and Maintain the AppleTalk Network

See the end of this chapter for configuration examples.

Enable AppleTalk Routing

To enable AppleTalk routing, first enable it on the communication server, then configure each interface for AppleTalk. These are the only two tasks you must perform when configuring AppleTalk routing.

To configure an interface for AppleTalk, assign an AppleTalk address or cable range to the interface and then assign one or more zone names to the interface. You can perform these tasks either manually or dynamically.

You also can enable our communication servers and routers to perform transition mode routing from nonextended to extended AppleTalk.

Enable AppleTalk Routing on the Communication Server

To enable AppleTalk routing on the communication server, perform the following task in global configuration mode:

Task	Command
Enable AppleTalk routing.	**appletalk routing**

Manually Configure an Interface

You can manually configure an interface for nonextended or extended AppleTalk routing.

To manually configure an interface for nonextended AppleTalk routing, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Assign an AppleTalk address to the interface.	**appletalk address** *network.node*
Step 2	Assign a zone name to the interface.	**appletalk zone** *zone-name*

After you assign the address and zone name, the interface will attempt to verify them with another operational router or communication server on the connected network. If there are any discrepancies, the interface will not become operational. If there are no neighboring operational routers or communication servers, the communication server will assume the interface's configuration is correct, and the interface will become operational.

To manually configure an interface for extended AppleTalk routing, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Assign a cable range to an interface.	**appletalk cable-range** *cable-range* [*network.node*]
Step 2	Assign a zone name to the interface.	**appletalk zone** *zone-name*

You can assign more than one zone name to a cable range. If you do so, the first name you assign is considered to be the default zone.

You can define up to 255 zones.

After you assign the address and zone names, the interface will attempt to verify them with another operational router or cummunciation server on the connected network. If there are any discrepancies, the interface will not become operational. If there are no neighboring operational routers or communication servers, the communication server will assume the interface's configuration is correct, and the interface will become operational.

Dynamically Configure an Interface

If a nonextended or an extended interface is connected to a network that has at least one other operational AppleTalk router or communication server, you can dynamically configure the interface using *discovery mode*. In discovery mode, an interface acquires information about the attached network from an operational router or communication server and then uses this information to configure itself.

Using discovery mode to configure interfaces saves time if the network numbers, cable ranges, or zone names change. If this happens, you need to make the changes only on one operational router or communication server.

Discovery mode is useful when you are changing a network configuration or when you are adding a communication server to an existing network.

Note that discovery mode does not run over serial lines.

If there is no operational router or communication server on the attached network, you must manually configure the interface as described in the previous sections. Also, if a discovery mode interface is restarted, another operational router or communication server must be present before the interface will become operational.

A nondiscovery-mode interface (also called a seed router) starts up as follows. The seed router acquires its configuration from memory. If the stored configuration is not completely specified when you assign an AppleTalk address to an interface or which you assign a cable range and a zone name to an interface the interface will not start up. If the stored configuration is completely specified, the interface will attempt to verify the stored configuration with another router or communication server on the attached network. If there is any discrepancy, the interface will not start up. If there are no neighboring operational routers communication servers, the communication server will assume the interface's stored configuration is correct, and the interface will become operational.

Using discovery mode does not affect an interface's ability to respond to configuration queries from other routers or communication servers on the connected network once the interface becomes operational.

When activating discovery mode, you do not need to assign a zone name. The interface will acquire the zone name from another interface.

Caution Do not enable discovery mode on all routers or communication servers on a network. If you do and they all restart simultaneously (for instance, after a power failure), the network will be inaccessible until you manually configure at least one router or communication server.

Dynamically Configure a Nonextended Interface

You can activate discovery mode on a nonextended interface in one of two ways, depending on whether you know the network number of the attached network.

In the first method, you immediately place the interface into discovery mode by specifying an AppleTalk address of 0.0. Use this method when you do not know the network number of the attached network. To use this method, perform the following task in interface configuration mode:

Task	Command
Place the interface into discovery mode by assigning it the AppleTalk address 0.0.	**appletalk address 0.0**

In the second method, you first assign an address to the interface and then explicitly enable discovery mode. Use this method when you know the network number of the attached network. Note, however, that you are not required to use this method when you know the network number. To use this method, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Assign an AppleTalk address to the interface.	**appletalk address** *network.node*
Step 2	Place the interface into discovery mode.	**appletalk discovery**

Dynamically Configure an Extended Interface

You can activate discovery mode on an extended interface in one of two ways, depending on whether you know the cable range of the attached network.

In the first method, you immediately place the interface into discovery mode by specifying a cable range of 0–0. Use this method when you do not know the network number of the attached network. To use this method, perform the following task in interface configuration mode:

Task	Command
Place the interface into discovery mode by assigning it the cable range 0-0.	**appletalk cable-range 0–0**

In the second method, you first assign cable ranges and then explicitly enable discovery mode. Use this method when you know the cable range of the attached network. Note, however, that you are not required to use this method if you know the cable range. To use this method, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Assign an AppleTalk address to the interface.	**appletalk cable-range** *cable-range* [*network***.***node*]
Step 2	Place the interface into discovery mode.	**appletalk discovery**

Configure Transition Mode

Our communication server can route packets between extended and nonextended AppleTalk networks that coexist on the same cable. This type of routing is referred to as transition mode.

To use transition mode, you must have two communication server ports connected to the same physical cable. One port is configured as a nonextended AppleTalk network, and the other port is configured as an extended AppleTalk network. Each port must have a unique network number, because you are routing between two separate AppleTalk networks: the extended network and the nonextended network.

To configure transition mode, you must have two ports on the same communication server that are connected to the same physical cable. You configure one port as a nonextended AppleTalk network by performing the following tasks in interface configuration mode:

Task		Command
Step 1	Assign an AppleTalk address to the interface.	**appletalk address** *network.node*
Step 2	Assign a zone name to the interface.	**appletalk zone** *zone-name*

You configure the second port as an extended AppleTalk network by performing the following tasks in interface configuration mode:

Task		Command
Step 1	Assign an AppleTalk cable range to the interface.	**appletalk cable-range** *cable-range* [*network.node*]
Step 2	Assign a zone name to the interface.	**appletalk zone** *zone-name*

When you enter interface configuration mode, the type of interface must be the same for both ports (for example, both could be Ethernet) and the interface number must be different (for example, 0 and 1).

Create an AppleTalk Routing Process

You can configure the RTMP or Enhanced IGRP routing protocols on any interface. You can also configure the AURP routing protocol on a tunnel interface. To create an AppleTalk routing process, perform the following task in interface configuration mode:

Task	Command
Create an AppleTalk routing process.	**appletalk protocol** {**aurp** | **eigrp** | **rtmp**}

Control Access to AppleTalk Networks

An *access list* is a list of AppleTalk network numbers or zones that is maintained by the communication server and used to control access to or from specific zones or networks.

The communication server supports two general types of AppleTalk access lists:

- AppleTalk-style access lists, which are based on AppleTalk zones
- IP-style access lists, which are based on network numbers

AppleTalk-style access lists regulate the internetwork using zone names. Zone names are good control points because they are the only network-level abstraction that users can access. You can express zones names either explicitly or by using generalized argument keywords. Thus, using AppleTalk access lists simplifies network management and allows for greater flexibility when adding segments, because reconfiguration requirements are minimal.

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

IP-style access lists control network access based on network numbers. This feature can be useful in defining access lists that control the disposition of networks that overlap, are contained by, or exactly match a specific network number range. One class of problem addressed by the use of IP-style access lists involves the potential assignment of conflicting network numbers to different networks. You can use an access list to restrict the network numbers and zones that a department can advertise, thereby limiting advertisement to an authorized set of networks. In general, AppleTalk-style access lists are insufficient for this application.

In general, however, using IP-style access lists is not recommended because the controls are not optimal: they ignore the logical mapping provided by AppleTalk zones. One problem with IP-style access lists is that when you add networks to a zone, you must reconfigure each secure router or communication server. Another problem is that because anyone can add network segments (for example, if one group of users gets a LaserWriter and installs a Cayman GatorBox, this creates a new network segment), the potential for confusion and misconfiguration is significant.

You can combine zone and network entries in a single access list. Network filtering is performed first, then zone filtering is applied to the result. However, for optimal performance, access lists should not include both zones (AppleTalk-style) and numeric network (IP-style) entries.

You can filter the following types of AppleTalk packets:

- Data packets
- Routing table updates
- GetZoneList (GZL) requests

Note The three types of filters are completely independent of each other. This means that if, for example, you apply a data packet filter to an interface, that filter has no effect on incoming routing table updates or GZL requests that pass through that interface. The exception to this is that outgoing routing update filters can affect GZL updates.

AppleTalk network access control differs from that of other protocols in that the order of the entries in an access list is not important. However, keep the following constraints in mind when defining access lists:

- You must design and type access list entries properly to ensure that entries do not overlap each other. An example of an overlap is if you were to enter a “permit network *xxx*” command and then enter a “deny network *xxx*” command. If you do enter entries that overlap, the last one you entered overwrites and removes the previous one from the access list. In the example earlier in this paragraph, this means that the “permit network” statement would be removed from the access list when you typed the “deny network” statement.
- Each access list always has a method for handling packets or routing updates that do not satisfy any of the access control statements in the access list.

To explicitly specify how you want these packets or routing updates to be handled, use the **access-list other-access** global configuration command when defining access conditions for networks and cable ranges, and use the **access-list additional-zones** global configuration command when defining access conditions for zones. If you use one of these commands, it does not matter where in the list you place it: The software automatically places the **access-list other-access** or **access-list additional-zones** command at the end of the access list. (With other protocols, you must type the equivalent commands last.)

If you do not explicitly specify how to handle packets or routing updates that do not satisfy any of the access control statements in the access list, the packets or routing updates are automatically denied access and, in the case of data packets, are discarded.

You perform the following tasks to control access to AppleTalk networks. These tasks are described in the sections that follow.

Step 1 Create access lists.

Step 2 Create filters.

Create Access Lists

An access list defines the conditions used to filter packets sent in to or out of the interface. Each access list is identified by a number. All **access-list** commands that specify the same access-list number create a single access list.

A single access list can contain any number and any combination of **access-list** commands. You can include network and cable range **access-list** commands and zone **access-list** commands in the same access list. However, you can specify only one each of the commands that specify default actions to take if none of the access conditions are matched. For example, a single access list can include only one **access-list other-access** command to handle networks and cable ranges that do not match the access conditions and only one **access-list additional-zones** command to handle zones that do not match the access conditions.

To create access lists that define access conditions for networks and cable ranges (IP-style access lists), perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a single network number.	**access-list** *access-list-number* {**deny** | **permit**} **network** *network*
Define access for a single cable range.	**access-list** *access-list-number* {**deny** | **permit**} **cable-range** *cable-range*
Define access for an extended or a nonextended network that overlaps any part of the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **includes** *cable-range*
Define access for an extended or a nonextended network that is included entirely within the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **within** *cable-range*
Define the default action to take for access checks that apply to network numbers or cable ranges.	**access-list** *access-list-number* {**deny** | **permit**} **other-access**

To create access lists that define access conditions for zones (AppleTalk-style access lists), perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a zone.	**access-list** *access-list-number* {**deny** | **permit**} **zone** *zone-name*
Define the default action to take for access checks that apply to zones.	**access-list** *access-list-number* {**deny** | **permit**} **additional-zones**

Create Filters

A filter examines specific packets that pass through an interface and permits or denies them based on the conditions defined in the access lists that have been applied to that interface.

You can filter the following types of AppleTalk packets:

- Data packets
- Routing table updates
- GetZoneList (GZL) requests

You can apply one of each type of filter to each interface, for a total of three filters per interface. Each filter can use the same access list or different access lists.

Data packet filters and incoming routing table update filters use access lists that define conditions for networks and cable ranges only. Outgoing routing update filters use access lists that define conditions for networks, cable ranges, and zones. GZL filters use access lists that define conditions for zones only.

The following sections explain the tasks for creating AppleTalk filters.

Create Data Packet Filters

A *data packet filter* checks data packets being sent out an interface. If the packets' source network or cable range has access denied, these packets are not transmitted but rather, are discarded.

Data packet filters use access lists that define conditions for networks and cable ranges only. They ignore any zone information that might be in the access list.

When you apply a data packet filter to an interface, you should ensure that all networks or cable ranges within a zone are governed by the same filters.

To create a data packet filter, perform the following tasks:

Step 1 Create a network-only access list.

Step 2 Apply a data packet filter to an interface.

To create a network-only access list, perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a single network number.	**access-list** *access-list-number* {**deny** | **permit**} **network** *network*
Define access for a single cable range.	**access-list** *access-list-number* {**deny** | **permit**} **cable-range** *cable-range*

Task	Command
Define access for an extended or a nonextended network that overlaps any part of the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **includes** *cable-range*
Define access for an extended or a nonextended network that is included entirely within the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **within** *cable-range*
Define the default action to take for access checks that apply to network numbers or cable ranges.	**access-list** *access-list-number* {**deny** | **permit**} **other-access**

To apply the data packet filter to an interface, perform the following task in interface configuration mode:

Task	Command
Apply the data packet filter to the interface.	**appletalk access-group** *access-list-number*

For an example of how to create data packet filters, see the section "AppleTalk Access List Examples" later in this chapter.

Create Routing Table Update Filters

Routing table update filters control which updates the local routing table accepts and which routes the local router or communication server advertises in its routing updates. You create distribution lists to control the filtering of routing updates.

Filters for incoming routing updates use access lists that define conditions for networks and cable ranges only. Filters for outgoing routing updates use access lists that define conditions for networks and cable ranges, and for zones.

When filtering incoming routing updates, each network number and cable range in the update is checked against the access list. If you have not applied an access list to the interface, all network numbers and cable ranges in the routing update are added to the routing table. If an access list has been applied to the interface, only network numbers and cable ranges that are not explicitly or implicitly denied are added to the routing table:

The following conditions are also applied when filtering routing updates generated by the local router or communicatiion server:

- The network number or cable range is not a member of a zone that is explicitly or implicitly denied.
- If partial zone processing is disabled (the default), the network number or cable range is not a member of a zone that contains other denied network numbers and/or cable ranges.

To create a filter for routing table updates received on an interface, perform the following tasks:

Step 1 Create an access list.

Step 2 Apply a routing table update filter to an interface.

To create an access list, perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a single network number.	**access-list** *access-list-number* {**deny** | **permit**} **network** *network*
Define access for a single cable range.	**access-list** *access-list-number* {**deny** | **permit**} **cable-range** *cable-range*
Define access for an extended or a nonextended network that overlaps any part of the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **includes** *cable-range*
Define access for an extended or a nonextended network that is included entirely within the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **within** *cable-range*
Define the default action to take for access checks that apply to network numbers or cable ranges.	**access-list** *access-list-number* {**deny** | **permit**} **other-access**

Caution Ensure that access lists used to filter incoming routing updates do not contain any zone entries. If they do, these entries might cause undefined behavior.

To apply the filter to incoming routing updates on an interface, perform the following task in interface configuration mode:

Task	Command
Apply the routing update filter.	**appletalk distribute-list** *access-list-number* **in**

For an example of how to create a filter for incoming routing table updates, see the section "AppleTalk Access List Examples."

To create a filter for routing table updates sent out on an interface, perform the following tasks:

Step 1 Create an access list.

Step 2 Apply a routing table update filter to an interface.

To create an access list, perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a single network number.	**access-list** *access-list-number* {**deny** | **permit**} **network** *network*
Define access for a single cable range.	**access-list** *access-list-number* {**deny** | **permit**} **cable-range** *cable-range*
Define access for an extended or a nonextended network that overlaps any part of the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **includes** *cable-range*

Task	Command
Define access for an extended or a nonextended network that is included entirely within the specified range.	**access-list** *access-list-number* {**deny** | **permit**} **within** *cable-range*
Define the default action to take for access checks that apply to network numbers or cable ranges.	**access-list** *access-list-number* {**deny** | **permit**} **other-access**
Define access for a zone.	**access-list** *access-list-number* {**deny** | **permit**} **zone** *zone-name*
Define the default action to take for access checks that apply to zones.	**access-list** *access-list-number* {**deny** | **permit**} **additional-zones**

To apply a filter to routing updates sent out on an interface, perform the following task in interface configuration mode:

Task	Command
Apply the routing update filter.	**appletalk distribute-list** *access-list-number* **out**

Create GetZoneList (GZL) Filters

The Macintosh Chooser uses ZIP GZL requests to compile a list of zones from which the user can select services. Any router or communication server on the same network as the Macintosh can respond to these requests with a GZL reply. You can create a GZL filter on the communication server to control which zones the communication server mentions in its GZL replies. This has the effect of controlling the list of zones that are displayed by the Chooser.

When defining GZL filters, you should ensure that all routers and communication servers on the same network filter GZL replies identically. Otherwise, the Chooser will list different zones depending upon which router or communication server responded to the request. Also, inconsistent filters can result in zones appearing and disappearing every few seconds when the user remains in the Chooser. Because of these inconsistencies, you should normally apply GZL filters only when all routers and communication servers in the internetwork are our routers or communication servers, unless the other vendors' routers have a similar feature.

When a ZIP GZL reply is generated, only zones that satisfy the following conditions are included:

- If partial zones are permitted, at least one network number or cable range that is a member of the zone is explicitly or implicitly permitted.
- If partial zones are not permitted (the default), all network numbers or cable ranges that are members of the zone are explicitly or implicitly permitted.
- The zone is explicitly or implicitly permitted.

Replies to GZL requests also are filtered by any outgoing routing update filter that has been applied to the same interface. You need to apply a GZL filter only 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.

Using a GZL filter is not a complete replacement for anonymous network numbers. In order to prevent users from seeing a zone, all routers and communication servers must implement the GZL filter. If there are any routers or communication servers on the network from other vendors, the GZL filter will not have a consistent effect.

To create a GZL filter, perform the following tasks:

Step 1 Create an access list.

Step 2 Apply a GZL filter to an interface.

To create an access list, perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a zone.	**access-list** *access-list-number* {**deny** | **permit**} **zone** *zone-name*
Define the default action to take for access checks that apply to zones.	**access-list** *access-list-number* {**deny** | **permit**} **additional-zones**

To apply the GZL filter to an interface, perform the following task in interface configuration mode:

Task	Command
Apply the GZL filter.	**appletalk getzonelist-filter** *access-list-number*

Enable ZIP Reply Filters

ZIP reply filters limit the visibility of zones from routers and communication servers in unprivileged regions throughout the internetwork. These filters filter the zone list for each network provided by a router or communication server to neighboring routers to remove restricted zones.

ZIP reply filters apply to downstream routers and communication servers, not to end stations on networks attached to the local communication server. With ZIP reply filters, when downstream routers and communication servers request the names of zones in a network, the local communication server replies with the names of visible zones only. It does not reply with the names of zones that have been hidden with a ZIP reply filter. To filter zones from end stations, use GZL filters.

To create a ZIP reply filter, perform the following tasks:

Step 1 Create an access list.

Step 2 Apply a ZIP reply filter to an interface.

To create an access list, perform one or both of the following tasks in global configuration mode:

Task	Command
Define access for a zone.	**access-list** *access-list-number* {**deny** | **permit**} **zone** *zone-name*
Define the default action to take for access checks that apply to zones.	**access-list** *access-list-number* {**deny** | **permit**} **additional-zones**

To apply the ZIP reply filter to an interface, perform the following task in interface configuration mode:

Task	Command
Apply the ZIP reply filter.	**appletalk zip-reply-filter** *access-list-number*

Enable Partial Zone Filters

If access to any network in a zone is denied, access to that zone is also denied by default. However, if you enable partial zones, access to other networks in that zone is no longer denied.

The permitting of partial zones provides IP-style access control. If enabled, the access control list behavior associated with prior software releases is restored. In addition, NBP cannot ensure consistency and uniqueness of name bindings.

If you permit partial zones, 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.

To enable partial zone filters, perform the following task in global configuration mode:

Task	Command
Permit access to networks in a zone in which access to another network in that zone is denied.	**appletalk permit-partial-zones**

Permitting partial zones affects the outgoing routing update and GZL filters.

Configure the Name Display Facility

The AppleTalk Name Binding Protocol (NBP) associates AppleTalk network entity names (that is, AppleTalk network-addressable services) with network addresses. NBP allows you to specify descriptive or symbolic names for entities instead of their numerical addresses. When you specify the name of an AppleTalk device, NBP translates the device's entity name into the device's network address. The name binding process includes name registration, name confirmation, name deletion, and name lookup.

Node addresses can change frequently because AppleTalk uses dynamic addresses. Therefore, NBP associates numerical node addresses with aliases that continue to reference the correct addresses if the addresses change. These node addresses do not change very frequently because each device keeps track of the last node number it was assigned. Typically, node numbers change only if a device is shut down for an extended period of time or if it is moved to another network segment.

To control the communication server's name display facility, perform one or both of the following tasks in global configuration mode:

Task	Command
Specify which service types are retained in the name cache.	**appletalk lookup-type** *service-type*
Set the interval between service pollings by the communication server on its AppleTalk interfaces.	**appletalk name-lookup-interval** *seconds*

Set up Special Configurations

To set up special configurations, perform the tasks in the following sections, as appropriate:

- Configure AURP
- Configure Free-Trade Zones
- Configure SNMP in AppleTalk Networks

- Configure AppleTalk Tunneling
- Configure AppleTalk MacIP
- Configure IPTalk

Configure AURP

The AppleTalk Update-based Routing Protocol (AURP) is a standard Apple Computer routing protocol that provides enhancements to the AppleTalk routing protocols that are compatible with AppleTalk Phase 2. The primary function of AURP is to connect two or more noncontiguous AppleTalk internets that are separated by a non-AppleTalk network, such as IP. In these configurations, you would want to use AURP instead of RTMP, because AURP broadcasts fewer data packets than RTMP.

You configure AURP on a tunnel interface. Tunneling encapsulates an AppleTalk packet inside an IP packet, which is sent across the backbone to a destination router or communication server. The destination router or communication server then extracts the AppleTalk packet and, if necessary, routes it to an AppleTalk network. The encapsulated packet benefits from any features normally applied to IP packets, including fragmentation, default routes, and load balancing.

To configure AURP, perform the following tasks:

Task	Command
Enable route redistribution.	**appletalk route-redistribution**
Configure an interface to be used by the tunnel.	**interface** *type number*
Configure an IP address.	**ip address** *ip-address*
Configure tunnel interface.	**interface tunnel** *type number*
Create an AURP routing process.	a**ppletalk protocol aurp**
Specify the interface out which the encapsulated packets will be sent.	**tunnel source** [*type number* | *ip-address*]
Specify the IP address of the communication server at the far end of the tunnel.	**tunnel destination** [*ip-address* | *hostname*]
Enable AURP tunneling.	**tunnel mode aurp**

By default, AURP sends routing updates every 30 seconds. To modify this interval, perform the following task in global configuration mode:

Task	Command
Set the minimum interval between AURP routing updates.	**appletalk aurp update-interval** *seconds*

To set the AURP last-heard-from timer value, perform the following task in interface configuration mode:

Task	Command
Set the AURP last-heard-from timer value.	**appletalk aurp tickle-time** *seconds*

Configure Free-Trade Zones

A free-trade zone is a part of an AppleTalk internetwork that is accessible by two other parts of the internetwork, neither of which can access the other. You might want to create a free-trade zone to allow the exchange of information between two organizations that otherwise want to keep their internetworks isolated from each other or that do not have physical connectivity with one another.

To establish a free-trade zone, perform the following task in interface configuration mode:

Task	Command
Establish a free-trade zone.	**appletalk free-trade-zone**

For an example of how to configure a free-trade zone, see the section "Hiding and Sharing Resources with Access List Examples," which contains the section "Establishing a Free-Trade Zone Example."

Configure SNMP in AppleTalk Networks

The Simple Network Management Protocol (SNMP) normally uses the User Datagram Protocol (UDP), IP's connectionless datagram service, to monitor network entities. Our software lets you run SNMP using DDP, the AppleTalk datagram service. Use DDP if you have SNMP consoles running on a Macintosh.

You must configure AppleTalk routing globally and on an interface basis before you configure SNMP for the communication server.

To configure SNMP in AppleTalk networks, perform the following tasks starting in global configuration mode:

Task		Command
Step 1	Disable SNMP on the communication server.	**no snmp server**
Step 2	Enable AppleTalk routing on the communication server.	**appletalk routing**
Step 3	Enable Appletalk event logging.	**appletalk event-logging**
Step 4	Enter interface configuration mode.	**interface** *interface-numbe*r
Step 5	Enable IP routing on the interface.	**ip address** *address*
Step 6	Enable AppleTalk routing on the interface.	**appletalk cable-range** *cable-range* [*network.node*]
Step 7	Set a zone name for the AppleTalk network.	**appletalk zone** *zone-name*
Step 8	Enable SNMP server operations.	**snmp-server community** *string* [**0**] [**0**]

For an example of configuring SNMP, see the section "SNMP Example."

For information about configuring SNMP on the communication server, refer to the "Managing the System" chapter.

Configure AppleTalk Tunneling

When connecting two AppleTalk networks with a non-AppleTalk backbone such as IP, the relatively high bandwidth consumed by the broadcasting of RTMP data packets can severly hamper the backbone's network performnace. You can solve this problem by tunneling AppleTalk through a

foreigh protocol, such as IP. Tunneling encapsulates an AppleTalk packet inside the foreign protocol packet, which is then sent across the backbone to a destination router or communication server. The destination router or communication server then de-encapsulates the AppleTalk packet and, if necessary, routes the packet to a normal AppleTalk network. Because the encapsulated AppleTalk packet is sent in a directed manner to a remote IP address, bandwidth usage is greatly reduced. Furthermore, the encapsulated packet benefits from any features normally enjoyed by IP packets, including default routes and load balancing.

There are two ways to tunnel AppleTalk. The first method implements Cayman tunneling as designed by Cayman Systems. This method enables communication servers to interoperate with Cayman GatorBoxes. The second method is a proprietary tunnel protocol known as generic routing encapsulation (GRE).

When you use Cayman tunneling, you can have our routers or communication servers at either end of the tunnel, or you can have a GatorBox at one end and our routers or communication server at the other end. When you use GRE tunneling, you must have only our routers or communication servers at both ends of the tunnel connection.

Multiple tunnels originating from the communication server are supported.

Logically, tunnels are point-to-point links. This requires that you configure a separate tunnel for each link.

To configure a Cayman tunnel, perform the following tasks in starting and global configuration mode:

Task		Command
Step 1	Configure a tunnel interface.	**interface tunnel** *number*
Step 2	Specify the interface out which the encapsulated packets will be sent.	**tunnel source** *interface*
Step 3	Specify the IP address of the router or communication server at the far end of the tunnel.	**tunnel destination** *interface*
Step 4	Enable Cayman tunneling.	**tunnel mode cayman**

Caution Do not configure a Cayman tunnel with an AppleTalk network address.

To configure a GRE tunnel, perform the following tasks in starting and global configuration mode:

Task		Command
Step 1	Configure a tunnel interface.	**interface tunnel** *number*
Step 2	Specify the interface out which the encapsulated packets will be sent.	**tunnel source** *interface*
Step 3	Specify the IP address of the router or communication server at the far end of the tunnel.	**tunnel destination** *interface*
Step 4	Enable Cayman tunneling.	**tunnel mode cayman**

Configure AppleTalk MacIP

Our communication servers implement MacIP, which is a protocol that allows routing of IP datagrams to IP clients using the DDP for low-level encapsulation.

Our communication servers implement the MacIP address management and routing services described in the draft Internet RFC, *A Standard for the Transmission of Internet Packets over AppleTalk Networks*. Our implementation of MacIP conforms to the September 1991 draft RFC with the following exceptions:

- Our communication servers do not fragment IP datagrams that exceed the DDP maximum transmission unit (MTU) and that are bound for DDP clients of MacIP.
- Our communication servers do not route to DDP clients outside of configured MacIP client ranges.

Some situations require the use of MacIP. For example, if some of your Macintosh users use AppleTalk Remote Access or are connected to the network using LocalTalk or PhoneNet cabling systems, then MacIP is required to provide access to IP network servers.

MacIP services also can be useful when you are managing IP address allocations for a large, dynamic Macintosh population. There are several advantages to using MacIP in this situation:

- Macintosh TCP/IP drivers can be configured in a completely standard way, regardless of the location of the Macintosh. Essentially, the dynamic properties of AppleTalk address management become available for IP address allocation.
- You can modify all global parameters, such as IP subnet mask, DNS services, and default routers or communication servers. Macintosh IP users receive the updates by restarting their local TCP/IP drivers.
- The network administrator can monitor MacIP address allocations and packet statistics remotely by using the Telnet application to attach to the communication server console. This allows central administration of IP allocations in remote locations. For Internet sites, it allows remote technical assistance.

However, there are several disadvantages in implementing MacIP on our communication servers:

- Each packet from a Macintosh client destined for an IP host or vice versa *must* pass through the communication server if the client is using the communication server as a MacIP server. The communication server is not always a necessary hop, so this increases traffic through the communication server. There is also a slight increase in communication server CPU use that is directly proportional to the number of packets delivered to and from active MacIP clients.
- Memory usage in the communication server increases in direct proportion to the total number of active MacIP clients (about 80 bytes per client).

To configure MacIP on our communication servers, AppleTalk must be configured as follows:

- AppleTalk routing must be enabled on at least one interface.
- IP routing must be enabled on at least one interface.
- The MacIP zone name you configure must be associated with a configured or seeded zone name.
- The MacIP server must reside in the default AppleTalk zone.
- Any IP address specified in configuring a MacIP server using an **appletalk macip** command must be *aliasable* to a specific IP interface on the communication server. Because the communication server is acting as a proxy for MacIP clients, you must use an IP address to which ARP can respond.

- 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 communication server 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).

When setting up MacIP routing, keep the following address range issues in mind:

- Static and dynamic resource statements are cumulative, and you can specify as many as necessary. However, if possible, you should specify a single all-inclusive range rather than several adjacent ranges. For example, specifying the range 131.108.121.1 to 131.108.121.10 is preferable to specifying the ranges 131.108.121.1 to 131.108.121.5 and 131.108.121.6 to 131.108.121.10.
- Overlapping resource ranges (for example, 131.108.121.1 to 131.108.121.5 and 131.108.121.5 to 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 command (such as **no appletalk macip dynamic** *ip-address ip-address* **zone** *server-zone*) to delete the original range, followed by the corrected range statement.
- You can add IP address allocations to a running server at any time as long as the new address range does not overlap with one of the current ranges.

To configure MacIP, perform the following tasks:

Step 1 Establish a MacIP server for a specific zone.

Step 2 Allocate IP addresses for Macintosh users by specifying at least one *dynamic* or *static* resource address assignment command for each MacIP server.

To establish a MacIP server for a specific zone, perform the following global configuration task:

Task	Command
Establish a MacIP server for a zone.	**appletalk macip server** *ip-address* **zone** *server-zone*

Note that the MacIP server must reside in the default AppleTalk zone.

You can configure multiple MacIP servers for a communication server, but you can assign only one MacIP server to a zone, and you can assign only one IP interface to a MacIP server. In general, you must be able to establish an alias between the IP address you assign with the **appletalk macip server** global configuration command and an existing IP interface. For implementation simplicity, the address you specify in this command should match an existing IP interface address.

A server is not registered by NBP until at least one MacIP resource is configured.

Dynamic clients are those that accept any IP address assignment within the dynamic range specified. Dynamic addresses are for users who do not require a fixed address, but can be assigned addresses from a pool.

To allocate IP addresses for Macintosh users if you are using dynamic addresses, perform the following global configuration task:

Task	Command
Allocate an IP address to a MacIP client.	**appletalk macip dynamic** *ip-address* [*ip-address*] **zone** *server-zone*

For an example of configuring MacIP with dynamic addresses, see the section “MacIP Examples” in this chapter.

Static addresses are for users who require fixed addresses for IP DNS services and for administrators who do not want addresses to change so that they always know the IP addresses of the devices on their network.

To allocate IP addresses for Macintosh users if you are using static addresses, perform the following global configuration task:

Task	Command
Allocate an IP address to be used by a MacIP client that has reserved a static IP address.	**appletalk macip static** *ip-address* [*ip-address*] **zone** *server-zone*

For an example of configuring MacIP with static addresses, see the section “MacIP Examples” in this chapter.

In general, it is recommended that you do not use fragmented address ranges in configuring ranges for MacIP. However, if this is unavoidable, use the **appletalk macip dynamic** command to specify as many addresses or ranges as required, and use the **appletalk macip static** command to assign a specific address or address range.

Configure IPTalk

IPTalk is a protocol for encapsulating AppleTalk packets in IP datagrams. IPTalk is used to route AppleTalk packets across non-AppleTalk backbones and to communicate with applications on hosts that cannot otherwise communicate via AppleTalk, such as the Columbia AppleTalk Package (CAP). IPTalk also allows serial connections to use IPTalk Serial Line Interface Protocol (SLIP) drivers.

If your system is a Sun or DEC ULTRIX system, it may be possible to run CAP directly in a mode that supports EtherTalk. In this case, your system would look 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.

The installation instructions for CAP refer to Kinetics IP (KIP) gateways and to the file *atalkatab.* If you use our IPTalk support, it is not necessary nor is it desirable to use *atalkatab.* Our IPTalk support assumes that you want to use the standard AppleTalk routing protocols to perform all wide-area AppleTalk routing. KIP and *atalkatab* are based on an alternative routing strategy in which AppleTalk packets are transmitted using IP routing. It is possible to use both strategies at the same time; however, the interaction between the two routing techniques is not well defined.

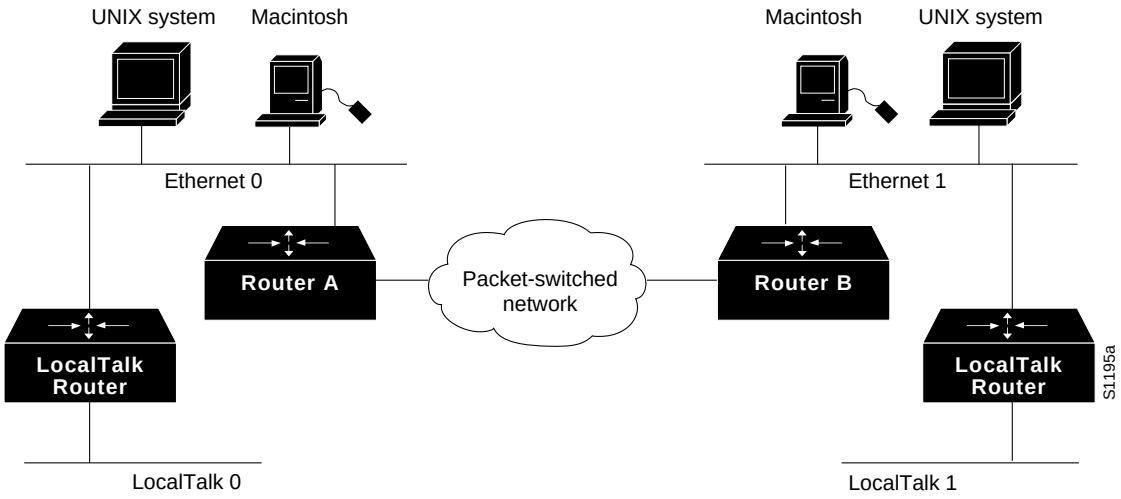
If your network has other vendors’ routers that support *atalkatab*, you should disable *atalkatab* support on them to avoid mixing the routing strategies. The installation instructions provided with some of these products encourage you to use *atalkatab* for complex networks. However, with our IOS software this is not necessary, because our implementation of IPTalk integrates IPTalk into the standard AppleTalk network routing.

The network diagram in Figure 17-1 illustrates how you should set up IPTalk. In this configuration, you enable both standard AppleTalk (EtherTalk) and IPTalk on the Ethernet networks on Communication Server A and Communication Server B. These servers then use EtherTalk to communicate with the LocalTalk servers and Macintosh computers, and IPTalk to communicate with the UNIX systems. On the LocalTalk communication servers, you also should enable both EtherTalk and IPTalk, making sure you configure IPTalk with *atalkatab* disabled. These communciation servers then use IPTalk to communicate with the UNIX systems adjacent to them and EtherTalk to communicate with the rest of the AppleTalk network. This configuration strategy minimizes the number of hops between communication servers. If you did not enable IPTalk on the

LocalTalk routers, systems on the LocalTalk router that wanted to communicate with the adjacent UNIX system would have to go through Router (or Communication Server) A or Router (or Communication Server) B. This creates an unnecessary extra hop.

Note In the configuration in Figure 17-1, all traffic between systems on the left and right sides of the packet-switched network transit via Routers (or CommServers) A and B using AppleTalk routing. If you were to enable *atalkatab* support on the LocalTalk routers, this would establish a hidden path between Routers (or CommServers) A and B, unknown to the standard AppleTalk routing protocols. In a large network, this could result in traffic taking inexplicable routes.

Figure 17-1 IPTalk Configuration Example



To configure IPTalk on an interface, perform the following tasks:

Step 1 Configure IP encapsulation of AppleTalk packets.

Step 2 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.

Configure IP Encapsulation of AppleTalk Packets

To allow AppleTalk to communicate with UNIX hosts running older versions of CAP that do not support native AppleTalk EtherTalk encapsulations, you need to configure IP encapsulation of AppleTalk packets. (Typically, Apple Macintosh users would communicate with these servers by routing their connections through a Kinetics FastPath router running KIP software.) Newer versions of CAP provide native AppleTalk EtherTalk encapsulations, so the IPTalk encapsulation is not longer required. Our implementation of IPTalk assumes that AppleTalk is already being routed on the backbone, because there is currently no LocalTalk hardware interface for our routers and communication servers.

You can configure IPTalk on an interface that already has a configured IP address.

Our implementation of IPTalk does not support manually configured AppleTalk-to-IP-address mapping. The address mapping provided is the same as the Kinetics IPTalk implementation when AppleTalk-to-IP-address mapping is not enabled. This address mapping works as follows: The IP subnet mask used on the communication server Ethernet interface on which IPTalk is enabled is

inverted (one's complement). The result is then masked against 255 (0xFF hexadecimal), and the result of this is then masked against the low-order 8 bits of the IP address to give the AppleTalk node number.

The following example configuration illustrates how the address mapping is done:

```
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
```

First, the IP subnet mask of 255.255.255.0 is inverted to give 0.0.0.255. This value then is masked with 255 to give 255. Next, 255 is masked with the low-order 8 bits of the interface IP address (118) to yield an AppleTalk node number of 118. This means that the AppleTalk address of the Ethernet 0 interface seen in the UDPZone zone is 30.118.

Note You should note the following caveat: If the host field of an IP subnet mask for an interface is longer than 8 bits, it will be possible to obtain conflicting AppleTalk node numbers. For instance, if the subnet mask for the Ethernet 0 interface above is 255.255.240.0, the host field is 12 bits wide.

To configure IP encapsulation of AppleTalk packets, perform the following tasks in interface configuration mode:

Task		Command
Step 1	Configure an IP address on the interface.	**ip address** *address* [*mask*]
Step 2	Enable IPTalk encapsulation on the interface.	**appletalk iptalk** *network***.***node zone*

For an example of configuring IPTalk, see the section "IPTalk Example" later in this chapter.

Specify the UDP Port Ranges

Implementations of IPTalk prior to April 1988 mapped well-known DDP socket numbers to privileged UDP ports starting at port number 768. In April 1988, the Network Information Center (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. Release 6 and later of the CAP program dynamically decides which port mapping to use. If there are no AppleTalk service entries in the UNIX system's */etc/services* file, CAP uses the older mapping starting at UDP port number 768.

The default UDP port mapping supported by our implementation of IPTalk is 768. If there are AppleTalk service entries in the UNIX system's */etc/services* file, you should specify the beginning of the UDP port mapping range.

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, perform the following task in global configuration mode:

Task	Command
Specify the starting UDP port number.	**appletalk iptalk-baseport**

For an example of configuring IPTalk, see the section "IPTalk Example."

Configure AppleTalk Control Protocol for PPP

You can configure an asynchronous interface on the access server to use AppleTalk Control Protocol (ATCP) so that users can access AppleTalk zones by dialing into the access server via PPP to this interface. This is done through a negotiation protocol, as defined in RFC 1378. Users accessing the network with ATCP can run AppleTalk and IP natively on a remote Macintosh, access any available AppleTalk zones from Chooser, use networked peripherals, and share files with other Macintosh users.

You create an internal network with the **appletalk internal-network** command. This is a virtual network and exists only for accessing an AppleTalk internet through the server.

To create a new AppleTalk zone, issue the **appletalk virtual-net** command and use a new zone name; this network number is then the only one associated with this zone. To add network numbers to an existing AppleTalk zone, use this existing zone name in the command; this network number is then added to the existing zone.

Routing is not supported on these interfaces.

To enable ATCP for PPP, perform the following tasks in interface configuration (asynchronous) mode:

Task		Command
Step 1	Define encapsulation as PPP on this interface.	**encapsulation ppp**
Step 2	Create an internal network on the server.	**appletalk virtual-net** *network-number zone-name*
Step 3	Enable client-mode on this interface.	**appletalk client-mode**

See the end of this chapter for "Configure AppleTalk Control Protocol for PPP Example."

Tune AppleTalk Network Performance

To tune AppleTalk network performance, you can perform one or more of the tasks described in the following sections:

- Control Routing Updates
- Assign Proxy Network Numbers
- Disable Checksum Generation and Verification
- Control the AppleTalk ARP Table
- Control the Delay between ZIP Queries
- Log Significant Network Events

- Disable Fast Switching

Control Routing Updates

The Routing Table Maintenance Protocol (RTMP) establishes and maintains the AppleTalk routing table. You can perform the following tasks to control packet routing and control routing updates:

- Disable the Processing of Routed RTMP Packets
- Disable the Transmission of Routing Updates
- Prevent the Advertisement of Routes to Networks with No Associated Zones
- Set Routing Table Update Timers

Disable the Processing of Routed RTMP Packets

By default, the communication server performs strict RTMP checking, which discards any RTMP packets sent by routers or communication servers that are not directly connected to the local router (that is, sent by routers that are not neighbors). This means that the local router does not accept any routed RTMP packets whose source is a remote network.

In almost all situations, you should leave RTMP checking enabled.

To disable RTMP checking and enable the processing of routed RTMP packets, perform the following task in global configuration mode:

Task	Command
Disable strict checking of RTMP updates.	**no appletalk strict-rtmp-checking**

Disable the Transmission of Routing Updates

By default, communication servers receive routing updates from their neighboring routers or communication servers and periodically send routing updates to their neighbors. You can configure a communication server so that it only receives routing updates, but does not send any updates. You might want to do this to keep a particular communication server from sending routing updates to its neighbors because it is unreliable.

To disable the transmission of routing updates, perform the following task in interface configuration mode:

Task	Command
Disable the transmission of routing updates on an interface.	**no appletalk send-rtmps**

Prevent the Advertisement of Routes to Networks with No Associated Zones

NBP uses ZIP to determine which networks belong to which zones. The communication server uses ZIP to maintain a table of the AppleTalk internetwork that maps network numbers to zone names.

By default, the communication server does not advertise routes to networks that have no associated zones. This prevents the occurrence of ZIP protocol storms, which can arise when corrupt routes are propagated and routers or communication servers broadcast ZIP requests to determine the network-zone associations. Not advertising routes to networks that do not have associated zones limits any ZIP protocol storms to a single network rather than allowing them to spread to the entire internetwork.

To allow the advertisement of routes to networks that have no associated zones, perform the following task in global configuration mode:

Task	Command
Allow the advertisement of routes to networks that have no associated zones.	**no appletalk require-route-zones**

The user zone lists can be configured to vary from interface to interface. However, this practice is discouraged because AppleTalk users expect to have the same user zone lists at any end node in the internetwork. This kind of filtering does not prevent explicit access via programmatic methods, but should be considered a user optimization whereby unused zones are suppressed. Use other forms of AppleTalk access control lists to actually secure a zone or network.

Set Routing Table Update Timers

The communication server sends routing table updates at regular intervals. In rare instances you might want to change this interval, such as when a communication server is busy and cannot send routing updates every 10 seconds, or when slower routers are incapable of processing received routing updates in a large network. If you do change the routing update interval, be sure to do so for all routers and communication servers on the network.

Caution Modifying the routing timers can degrade or destroy AppleTalk network connectivity. Many other AppleTalk router vendors provide no facility for modifying their routing timers, so adjusting our communication server's AppleTalk timers such that routing updates do not arrive at these other routers within the normal interval might result in loss of information about the network or loss of connectivity.

To change the routing table update timers, perform the following task in global configuration mode:

Task	Command
Change the routing update timers.	**appletalk timers** *update-interval valid-interval invalid-interval*

Assign Proxy Network Numbers

It is possible to have an AppleTalk internetwork in which some routers support only nonextended AppleTalk and others support only extended AppleTalk. You can enable interoperability between these two types of AppleTalk networks by assigning a proxy network number for each zone in which there is a router that supports only nonextended AppleTalk.

To assign proxy network numbers, perform the following task in global configuration mode:

Task	Command
Assign a proxy network number for each zone in which there is a router that supports only nonextended AppleTalk.	**appletalk proxy-nbp** *network-number zone-name*

For an example of configuring proxy network numbers, see the section "Assign Proxy Network Numbers."

Caution Do not also assign the proxy network number to a router or to a physical network.

You must assign one proxy network number for each zone. You can optionally define additional proxies with different network numbers to provide redundancy. Each proxy network number generates one or more packets for each forward request it receives, but discards all other packets sent to it. Thus, defining redundant proxy network numbers increases the NBP traffic linearly.

Disable Checksum Generation and Verification

By default, the communication server generates and verifies checksums for all AppleTalk packets (except routed packets). You might want to disable checksum generation and verification if you have older devices, such as LaserWriter printers, that cannot receive packets with checksums.

To disable checksum generation and verification, perform the following task in global configuration mode:

Task	Command
Disable the generation and verification of checksums for all AppleTalk packets.	**no appletalk checksum**

Control the AppleTalk ARP Table

You can perform the following tasks to control the AppleTalk ARP table:

- Set the timeout for ARP table entries
- Specify the time interval between the retransmission of ARP packets
- Specify the number of ARP retransmissions
- Disable the gleaning of ARP information from incoming packets

By default, entries in the AppleTalk ARP table are removed from the table if no update has been received in the last 4 hours. To change the ARP timeout interval, perform the following task in interface configuration mode:

Task	Command
Set the timeout for ARP table entries.	**appletalk arp-timeout** *interval*

AppleTalk ARP associates AppleTalk network addresses with media (data link) addresses. When AppleTalk must send a packet to another network node, the protocol address is passed to AppleTalk ARP, which undertakes a series of address negotiations to associate the protocol address with the media address.

If your AppleTalk network has devices that respond slowly, such as printers and overloaded file servers, you can lengthen the interval between AppleTalk ARP packets in order to allow the responses from these devices to be received. To do this, perform one or both of the following tasks in global configuration mode:

Task	Command
Specify the time interval between retransmission of ARP packets.	**appletalk arp** [**probe** | **request**] **interval** *interval*

Task	Command
Specify the number of retransmissions that will occur before abandoning address negotiations and using the selected address.	**appletalk arp** [**probe** | **request]** **retransmit-count** *number*

The communication server automatically derives ARP table entries from incoming packets. This process is referred to as "gleaning." Gleaning speeds up the process of populating the ARP table. To disable the gleaning of ARP table entries, perform the following task in interface configuration mode:

Task	Command
Disable the gleaning of ARP information from incoming packets.	**no appletalk glean-packets**

Control the Delay between ZIP Queries

By default, the communication server sends ZIP queries every 10 seconds and uses the information received to update its zone table. To change the ZIP query interval, perform the following task in global configuration mode:

Task	Command
Set the ZIP query interval.	**appletalk zip-query-interval** *interval*

Log Significant Network Events

You can log information about significant network events performed on the communication server, including routing changes, zone creation, port status, and address. To do this, perform the following task in global configuration mode:

Task	Command
Log significant events.	**appletalk event-logging**

Disable Fast Switching

Fast switching allows higher throughput by switching a packet using a cache created by previous packets. Fast switching is enabled by default on all interfaces that support fast switching.

Packet transfer performance is generally better when fast switching is enabled. However, you may want to disable fast switching in order to save memory space on interface cards and to help avoid congestion when high-bandwidth interfaces are writing large amounts of information to low-bandwidth interfaces.

To disable AppleTalk fast switching on an interface, perform the following task in interface configuration mode:

Task	Command
Disable AppleTalk fast switching.	**no appletalk route-cache**

Configure AppleTalk Enhanced IGRP

Enhanced IGRP is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco Systems, Inc. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of Enhanced IGRP have improved significantly over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all routers and communication servers involved in a topology change to synchronize at the same time. Routers and communication servers that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

Cisco's Enhanced IGRP Implementation

AppleTalk Enhanced IGRP provides the following features:

- Automatic redistribution. By default, AppleTalk RTMP routes are automatically redistributed into Enhanced IGRP, and AppleTalk Enhanced IGRP routes are automatically redistributed into RTMP. If desired, you can turn off redistribution. You also can completely turn off AppleTalk Enhanced IGRP and AppleTalk RTMP on the communication server or on individual interfaces.
- Configuration of routing protocols on individual interfaces.You can configure interfaces that have been configured for AppleTalk to use either RTMP, Enhanced IGRP, or both routing protocols. If two neighboring routers and/or communication servers are configured to used both RTMP and Enhanced IGRP, the Enhanced IGRP routing information will supersede the RTMP information. However, both routers and/or communciation servers will continue to send RTMP routing updates. This feature allows you to control the excessive bandwidth usage of RTMP on WAN links. Because a WAN link is a point-to-point link, there are no other devices on the link, and hence, there is no need to run RTMP to perform end-node router discovery. Using Enhanced IGRP on WAN links allows you to save bandwidth and, in the case of PSDNs, traffic charges.

Enhanced IGRP offers the following features:

- Fast convergence. The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- Partial updates. Enhanced IGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for Enhanced IGRP packets.
- Less CPU usage than IGRP. This occurs because full update packets do not have to be processed each time they are received.
- Neighbor discovery mechanism. This is a simple hello mechanism used to learn about neighboring routers or communication servers. It is protocol-independent.
- Scaling. Enhanced IGRP scales to large networks.

Enhanced IGRP has four basic components:

- Neighbor discovery/recovery
- Reliable transport protocol
- DUAL finite state machine
- Protocol-dependent modules

Neighbor discovery/recovery is the process that routers and communication servers use to dynamically learn of other routers and communication servers on their directly attached networks. Routers and communication servers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery/recovery is achieved with low overhead by periodically sending small hello packets. As long as hello packets are received, a router and communication server can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring routers can exchange routing information.

The reliable transport protocol is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some Enhanced IGRP packets must be transmitted reliably and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, such as Ethernet, it is not necessary to send hellos reliably to all neighbors individually. Therefore, Enhanced IGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets, such as updates, require acknowledgment, and this is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.

The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information, known as a metric, to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring router or communication server used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Even though the recomputation is not processor intensive, it is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, it will use any it finds in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. It is also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information received. Enhanced IGRP asks DUAL to make routing decisions, but the results are stored in the AppleTalk routing table. Also, Enhanced IGRP is responsible for redistributing routes learned by other AppleTalk routing protocols.

Enhanced IGRP Configuration Task List

To configure AppleTalk Enhanced IGRP, complete the tasks in the following sections. At a minimum, you must create the AppleTalk Enhanced IGRP routing process. The remaining tasks are optional.

- Enable AppleTalk Enhanced IGRP
- Configure Miscellaneous Parameters

Enable AppleTalk Enhanced IGRP

To create an AppleTalk Enhanced IGRP routing process, perform the following tasks:

Task		Command
Step 1	Enable an AppleTalk Enhanced IGRP routing process in global configuration mode.	**appletalk routing eigrp** *router-number*
Step 2	Enable Enhanced IGRP on an interface in interface configuration mode.	**appletalk protocol eigrp**

For an example of how to enable AppleTalk Enhanced IGRP, see the section "AppleTalk Enhanced IGRP Example."

To associate multiple networks with an AppleTalk Enhanced IGRP routing process, you can repeat this task.

Configure Miscellaneous Parameters

To configure miscellaneous AppleTalk Enhanced IGRP parameters, perform one or more of the following tasks:

- Disable Redistribution of Routing Information
- Adjust the Interval between Hello Packets and the Hold Time
- Disable Split Horizon

Disable Redistribution of Routing Information

By default, the communication server redistributes AppleTalk RTMP routes into AppleTalk Enhanced IGRP, and vice versa. Internal Enhanced IGRP routes are always preferred over external Enhanced IGRP routes. This means that if there are two Enhanced IGRP paths to a destination, the path that originated within the Enhanced IGRP autonomous sytem always will be preferred over the Enhanced IGRP path that originated from outside of the autonomous system, regardless of the metric. Redistributed RTMP routes always are advertised in Enhanced IGRP as external.

To disable route redistribution, perform the following task in global configuration mode:

Task	Command
Disable redistribution of RTMP routes into Enhanced IGRP and Enhanced IGRP routes into RTMP.	**no appletalk route-redistribution**

Adjust the Interval between Hello Packets and the Hold Time

You can adjust the interval between hello packets and the hold time.

Routers and communication servers periodically send hello packets to each other to dynamically learn of other routers and communciation servers on their directly attached networks. The communication servers use this information to discover who their neighbors are and to learn when their neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

You can configure the hold time, in seconds, on a specified interface for the AppleTalk Enhanced IGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds.

On very congested and large networks, 15 seconds may not be sufficient time for all routers and communication servers to receive hello packets from their neighbors. In this case, you may want to increase the hold time.

Note Do not adjust the hold time without advising technical support.

To change the interval between hello packets and the hold time, perform the following task in interface configuration mode:

Task	Command
Set the interval between hello packets and the hold time.	**appletalk eigrp-timers** *hello-interval hold-time*

Disable Split Horizon

Split horizon controls the sending of AppleTalk Enhanced IGRP update and query packets. When split horizon is enabled on an interface, these packets are not sent to destinations for which this interface is the next hop. This reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks information about routes from being advertised by a communication server out any interface from which that information originated. This behavior usually optimizes communication among multiple routers and communication servers, particularly when links are broken. However, with nonbroadcast networks, such as Frame Relay and SMDS, situations can arise for which this behavior is less than ideal. For these situations, you may wish to disable split horizon.

To disable split horizon, perform the following task in interface configuration mode:

Task	Command
Disable split horizon.	**no appletalk eigrp-splithorizon**

Configure AppleTalk Inter-Enterprise Routing

AppleTalk inter-enterprise routing provides support for AppleTalk internets, or domains. AppleTalk inter-enterprise routing allows two or more AppleTalk domains to be connected through a domain router in a manner that allows the resolution of conflicting AppleTalk network numbers (cable ranges) from different domains and hop-count reduction between domains.

An AppleTalk domain is a group of AppleTalk networks or cable ranges that are connected and that have the following characteristics:

- Each network number or cable range within a domain is unique within that domain.
- Each domain is separated from another domain by a domain router or communication server.
- There is no physical or virtual connection between the two AppleTalk domains other than through a domain router or communication server.

The domain router or communication server uses split horizon based on the domain. This means that routes from one domain are not advertised on other interfaces within this domain. Rather, only routes from a different domain are advertised out the interface.

AppleTalk inter-enterprise routing provides the following features:

- Network remapping allows you to remap remote network numbers to resolve numbering conflicts with network numbers on the local network segment.
- Hop-count reduction allows the creation of larger internetworks.
- Loop detection avoids having multiple routing table entries to the same remote network segment (domain).
- Support for fast switching for networks that have been remapped or on which hop-count reduction has been configured.

Note that only one domain router or communication server can separate two domains. That is, you cannot have two or more domain routers or communication servers to create redundant paths between domains. You can however establish redundant paths between domains by connecting them through more than one interface on the domain router or communication server that separates them. Figure 17-2 illustrates this configuration. In this figure, one domain router or communication server separates domains A and B. Two of the router's interfaces are in domain A (Ethernet interfaces 3 and 4), and three are in domain B (Ethernet interfaces 0, 1, and 2), thus providing redundant connections between the domains. Figure 17-3 illustrates an improper configuration. This configuration will create adverse effects, because domains A and B are connected by two domain routers.

Figure 17-2 Allowed Configuration of Domain Router Connecting Two Domains

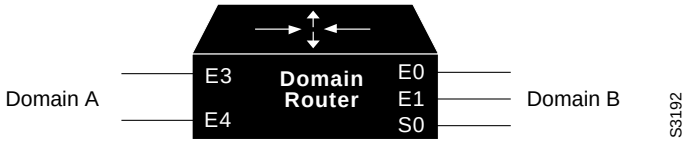
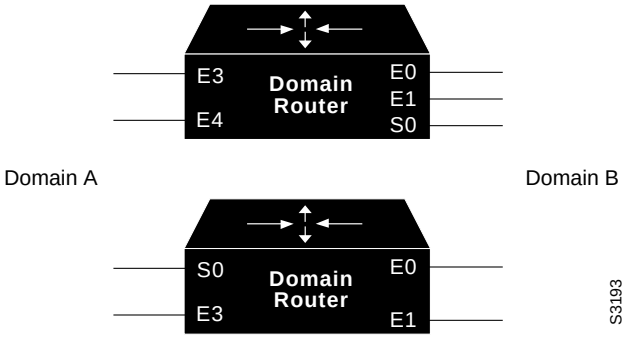


Figure 17-3 Improper Configuration of Domain Routers Connecting Two Domains



You can configure AppleTalk inter-enterprise routing only on routers and communciation servers running RTMP or Enhanced IGRP.

To configure AppleTalk inter-enterprise routing, you must perform the tasks described in the following section:

- Enable AppleTalk Inter-Enterprise Routing

Optionally, you can also perform the tasks described in the following sections:

- Remap Network Numbers
- Control Hop Count

Enable AppleTalk Inter-Enterprise Routing

To enable AppleTalk inter-enterprise routing, perform the following steps:

Step 1 Enable AppleTalk inter-enterprise routing on the communication server.

Step 2 Enable AppleTalk inter-enterprise routing on an interface.

To enable AppleTalk inter-enterprise routing on the communication server, perform the following task in global configuration mode:

Task	Command
Create a domain and assign it a name and number.	**appletalk domain** *domain-number* **name** *domain-name*

To enable AppleTalk inter-enterprise routing on an interface, perform the following task in interface configuration mode:

Task	Command
Assign a predefined domain number to an interface.	**appletalk domain-group** *domain-number*

Remap Network Numbers

When connecting two AppleTalk networks, there can be a conflict between network numbers or cables ranges on one network and those on the other. You can avoid conflicts by remapping the remote network's network numbers or cable ranges.

Each domain can have two domain mapping ranges to which to remap all incoming or outgoing network numbers or cable ranges.

To remap the network numbers or cable ranges on inbound packets, perform the following task in global configuration mode:

Task	Command
Remap packets inbound to the domain.	**appletalk domain** *domain-number* **remap-range in** *start-range-end-range*

To remap the network numbers or cable ranges on outbound packets, perform the following task in global configuration mode:

Task	Command
Remap packets outbound from the domain.	**appletalk domain** *domain-number* **remap-range out** *start-range-end-range*

Control Hop Count

When you join AppleTalk network segments to create domains, the distance across the combined internetworks is likely to exceed 15 hops, which is the maximum number of hops supported by RTMP. You can extend the network topology by configuring the communication server to reduce the hop-count value of packets that traverse it.

Reducing the hop-count value allows an AppleTalk router or communcation server to control the hop-count field in DDP packets and thus ensures that the packet reaches its final AppleTalk destination. Hop-count reduction allows the communication server to bypass the limitation of 16 hops before aging out packets. This feature is supported only on communication servers configured for AppleTalk Enhanced IGRP.

To enable hop-count reduction on the communication server, perform the following task in global configuration mode:

Task	Command
Enable hop-count reduction on the communication server.	**appletalk domain** *domain-number* **hop-reduction**

Configure AppleTalk over WANs

You can configure AppleTalk over dial-on-demand routing (DDR), Frame Relay, Switched Multimegabit Data Service (SMDS), and X.25 networks. To do this, configure the address mappings as described in the appropriate chapters for each protocol.

To use AppleTalk over DDR, you must define AppleTalk static routes. To do this, perform one of the following tasks in global configuration mode:

Task	Command
Define a static route on an extended AppleTalk network.	**appletalk static cable cable-range to** *network.node* **zone** *zone-name*
Define a static route on a nonextended AppleTalk network.	**appletalk static net** *network-number* **to** *network.node* **zone** *zone-name*

You can configure static routes only on interfaces that support DDR.

For an example of how to configure AppleTalk over DDR, see the section "AppleTalk over DDR Example" later in this chapter.

For X.25, you can configure only a nonextended AppleTalk network. Logically, this network is the same as a LocalTalk network, because both are *always* nonextended networks. All AppleTalk nodes within an X.25 network must be configured with the same AppleTalk network number. Also, the network numbers and zone names on both sides of the serial link must be the same. When mapping the AppleTalk address to the X.121 address of the communication server with the **x25 map** command, include the keyword **broadcast** to simulate the AppleTalk broadcast capability. This is necessary because X.25 does not support broadcasts, but AppleTalk does. The broadcast simulation is done as follows: If the broadcast flag is set, whenever a broadcast packet is sent, each X.121 address specified will receive it.

Monitor and Maintain the AppleTalk Network

The IOS software provides several commands you can use to monitor and maintain an AppleTalk network. In addition, you can use network monitoring packages, such as Apple Computer's *Inter•Poll*, to verify that a communication server is configured and operating properly. Use the commands described in this section to monitor an AppleTalk network using both communication server commands and network monitoring packages.

Monitor and Maintain the AppleTalk Network Using Communication Server Commands

To monitor and maintain the AppleTalk network, perform one or more of the following tasks at the EXEC prompt:

Task	Command
Enable recognition of pre-FDDITalk packets.	**appletalk pre-fdditalk**
Delete entries from the AppleTalk ARP (AARP) table.	**clear appletalk arp** [*network*.*node*]
Delete entries from the neighbor table.	**clear appletalk neighbor** [*neighbor-address* | *all*]
Delete entries from the routing table.	**clear appletalk route** *network*
Reset AppleTalk traffic counters.	**clear appletalk traffic**
Diagnose basic AppleTalk network connectivity (user-level command).	**ping appletalk** *network*.*node*
Diagnose basic AppleTalk network connectivity (privileged command).	**ping** [**appletalk**] [*network*.*node*]
Display the AppleTalk access lists currently defined.	**show appletalk access-lists**
Display the routes to networks that are directly connected or that are one hop away.	**show appletalk adjacent-routes**
List the entries in the AppleTalk ARP table.	**show appletalk arp**
Display pending events in the AppleTalk AURP update-events queue.	**show appletalk aurp events**
Display entries in the AURP private path database.	**show appletalk aurp topology**
Display the contents of the AppleTalk fast-switching cache.	**show appletalk cache**

Task	Command
List the neighbors discovered by AppleTalk Enhanced IGRP.	**show appletalk eigrp neighbors** [*interface*]
Display the contents of the AppleTalk Enhanced IGRP topology table.	**show appletalk eigrp topology** [*network-number* | **active** | **zero-successors**]
Display information about the communication server's AppleTalk internetwork and other parameters.	**show appletalk globals**
Display AppleTalk-related interface settings.	**show appletalk interface** [**brief**] [*type number*]
Display the status of all known MacIP clients.	**show appletalk macip-clients**
Display the status of a communication server's MacIP servers.	**show appletalk macip-servers**
Display statistics about MacIP traffic.	**show appletalk macip-traffic**
Display a list of NBP services offered by nearby routers and by other devices that support NBP.	**show appletalk name-cache**
Display the contents of the NBP name registration table.	**show appletalk nbp**
Display information about the AppleTalk routers directly connected to any network to which the communication server is directly connected.	**show appletalk neighbors** [*neighbor-address*]
Display domain remapping information.	**show appletalk remap** [**domain** *domain-number* [{**in** | **out**} [{**to** | **from**} *domain-network*]]]
Display the contents of the AppleTalk routing table.	**show appletalk route** [*network* | *type number*]
Display the process-level operations in all sockets in an interface.	**show appletalk sockets** [*socket-number*]
Displayed the defined static routes.	**show appletalk static**
Display the statistics about AppleTalk protocol traffic, including MacIP traffic.	**show appletalk traffic**
Display the contents of the zone information table.	**show appletalk zone** [*zone-name*]

Monitor the AppleTalk Network Using Network Monitoring Packages

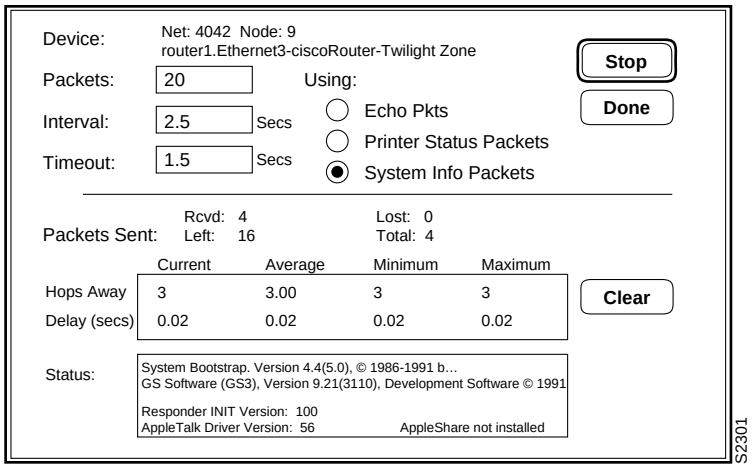
The IOS software supports network monitoring packages, such as Apple Computer's *Inter•Poll*, which are tools that use the AppleTalk responder and listener for verifying the communication server's configuration and operation. The communication server answers Appletalk responder request packets. These request packets are received by the listener, which is installed on the Appletalk interface name registration socket. The responder request packets include the bootstrap firmware version string, followed by the communication server operating software version string. These strings are displayed in the Macintosh System version and the Macintosh printer driver version fields, respectively, in applications such as Apple's *Inter•Poll.* The response packet contains strings similar to those displayed by the **show version** EXEC command.

The communication server returns the following information in response to responder request packets:

- System bootstrap version (ROM version).
- Software version.
- AppleTalk version—This is always 56, which is the first Apple Macintosh version that contained AppleTalk Phase 2 support.
- AppleTalk responder version—This is always 100, which indicates support of Version 1.0 responder packets.
- AppleShare status—This is reported as "not installed."

Figure 17-4 illustrates a typical output display for *Inter•Poll* that lists this information.

Figure 17-4 *Inter•Poll* Output



AppleTalk Configuration Examples

Use the following configuration examples in the following sections to help you configure AppleTalk routing on your communication server:

- Extended AppleTalk Network Example
- Nonextended AppleTalk Network Example
- Transition Mode Example
- AppleTalk Access List Examples
- GZL and ZIP Reply Filter Examples
- Hiding and Sharing Resources with Access List Examples
- MacIP Examples
- SNMP Example
- Proxy Network Number Example
- AppleTalk Enhanced IGRP Example

- AppleTalk over DDR Example
- IPTalk Example
- Configure AppleTalk Control Protocol for PPP Example

Extended AppleTalk Network Example

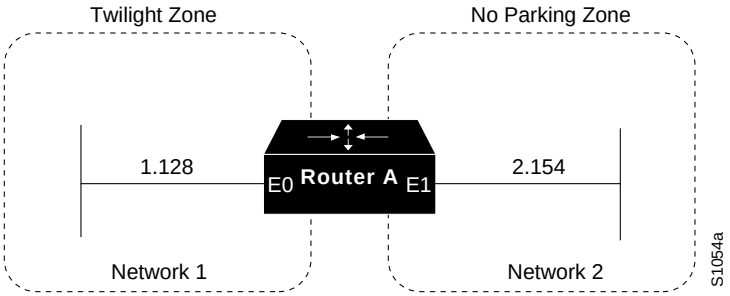
The following example configures an extended AppleTalk network. It defines the zones Purgatory and Underworld. The cable range of one allows compatibility with nonextended AppleTalk networks.

```
appletalk routing
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Purgatory
appletalk zone Underworld
```

Nonextended AppleTalk Network Example

The following example configures a nonextended AppleTalk network that allows routing between two Ethernet networks. Ethernet interface 0 is connected to network 1 at node 128, and Ethernet interface 1 is connected to network 2 at node 154. Network 1 is in the Twilight zone, and network 2 is in the No Parking zone. See Figure 17-5.

Figure 17-5 Nonextended AppleTalk Routing between Two Ethernet Networks

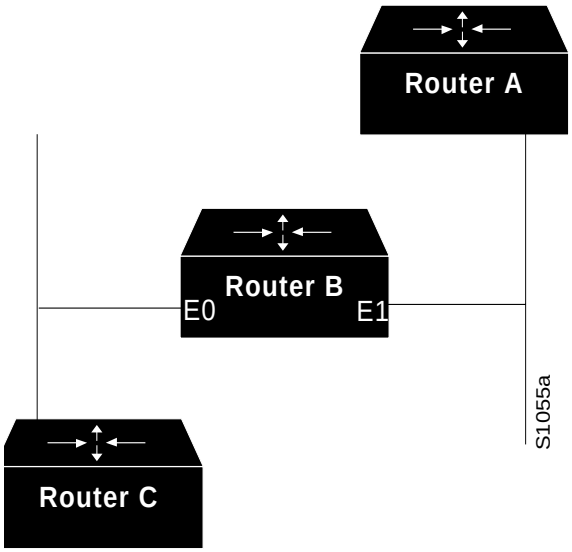



```
appletalk routing
!
interface ethernet 0
appletalk address 1.128
appletalk zone Twilight
!
interface ethernet 1
appletalk address 2.154
appletalk zone No Parking
```

Nonextended Network in Discovery Mode Example

The following example configures a nonextended network in discovery mode. There are seed routers on both networks to provide the zone and network number information to the interfaces when they start. Router (Communication Server) A supplies configuration information for Ethernet interface 1, and Router (Communication Server) C supplies configuration information for Ethernet interface 0. See Figure 17-6.

Figure 17-6 Routing in Discovery Mode



Use the following commands to configure this nonextended network in discovery mode:

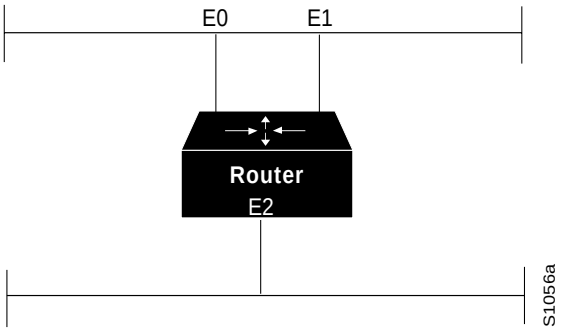
```
appletalk routing
!
interface ethernet 0
appletalk address 0.0
!
interface ethernet 1
appletalk address 0.0
```

Transition Mode Example

When in transition mode, the communication server can route packets between extended and nonextended AppleTalk networks that exist on the same cable.

To configure transition mode, you must have two ports connected to the same physical cable. One port is configured as a nonextended AppleTalk network, and the other is configured as an extended AppleTalk network. Both ports must have unique network numbers, because they are two separate networks. Figure 17-7 shows an example of the topology of this configuration.

Figure 17-7 Transition Mode Topology and Configuration



Use the following commands to configure this network. Note that networks 2-2 and 4-4 must have a cable range of one and a single zone in their zone lists. This is required to maintain compatibility with the nonextended network, network 3.

```
!This is an extended network.
interface ethernet 0
appletalk cable-range 2-2
appletalk zone No Parking
!
!This is a nonextended network.
interface ethernet 1
appletalk address 3.128
appletalk zone Twilight
!
!This is an extended network.
interface ethernet 2
appletalk cable-range 4-4
appletalk zone Do Not Enter
```

AppleTalk Access List Examples

Our implementation of AppleTalk provides several access list methods to control access to AppleTalk networks. The examples that follow illustrate these methods and show different approaches in applying access lists.

Defining an Access List to Filter Data Packets

The following commands create access list 601:

```
!Permit packets to be routed from network 55.
access-list 601 permit network 55

!Permit packets to be routed from network 500.
access-list 601 permit network 500

!Permit packets to be routed from networks 900 through 950.
access-list 601 permit cable-range 900-950

!Do not permit packets to be routed from networks 970 through 990.
access-list 601 deny includes 970-990

!Do not permit packets to be routed from networks 991 through 995.
access-list 601 permit within 991-995

!Deny routing to any network and cable range not specifically enumerated.
access-list 601 deny other-access
```

To use access list 601 to filter data packets, you apply it an interface (for example, Ethernet interface 0) using the following commands:

```
appletalk routing
interface ethernet 0
appletalk cable-range 50-50
appletalk zone No Parking
appletalk access-group 601
```

The following examples illustrate how Ethernet interface 0 would handle outgoing data packets:

- Packets sourced from cable range 50–50 would be permitted.
- Packets sourced from any network in the cable range 972–980 are denied because they explicitly match the **access-list deny includes 970-990** command.

Defining an Access List to Filter Incoming Routing Table Updates

The following commands create access list 602. This example illustrates how packets are processed by access lists; you would probably never create such a redundant access list.

```
access-list 602 permit network 55
access-list 602 permit cable 55-55
access-list 602 permit includes 55-55
access-list 602 permit within 55-55
```

To use this access list to filter routing table updates received on Ethernet interface 0, apply it to the interface using the following commands:

```
appletalk routing
interface ethernet 0
appletalk cable-range 55-55
appletalk zone No Parking
appletalk distribute-list 602 in
```

The following tables illustrate the process for accepting or rejecting routing update information. If the outcome of a test is true, the condition passes the access list specification and the **distribute-list** command specification is then applied.

Routing updates from network 55 would be processed as follows:

Access List Command	Outcome of Test
access-list 602 permit network 55	True
access-list 602 permit cable range 55-55	False
access-list 602 permit includes 55-55	True
access-list 602 permit within 55-55	True

Routing updates from cable range 55-55 would be processed as follows:

Access List Command	Outcome of Test
access-list 602 permit network 55	False
access-list 602 permit cable range 55-55	True
access-list 602 permit includes 55-55	True
access-list 602 permit within 55-55	True

Routing updates from cable range 55-55 would be processed as follows:

Access List Command	Outcome of Test
access-list 602 permit network 55	False
access-list 602 permit cable-range 55-55	False
access-list 602 permit includes 55-55	True
access-list 602 permit within 55-55	False

Comparison of Alternative Segmentation Solutions

With the flexibility allowed by our 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 two solutions to a particular problem and point out the inherent advantages of using AppleTalk-style access lists.

Consider a situation in which a company wants to permit customers to have direct access to several corporate file servers. Access is to be permitted to all devices in the zones named MIS and Corporate, but access is restricted to the Engineering zone because the file servers in these zones contain sensitive information. The solution is to create the appropriate access lists to enforce these access policies.

The company's AppleTalk internetwork consists of the following networks and zones:

Zone	Network Number or Cable Range
Engineering	69–69 3 4160–4160 15
MIS	666–777
Corporate	70–70 55 51004 4262–4262
World	88–88 9 9000–9999 (multiple networks exist in this range)

The communication server named Gatekeeper is placed between the World zone and the various company-specific zones. There can be an arbitrary number of routers or communication servers on either side of Gatekeeper. An Ethernet backbone exists on each side of Gatekeeper, connecting these other routers to Gatekeeper. On the router Gatekeeper, Ethernet interface 0 connects to the World backbone and Ethernet interface 1 connects to the Corporate backbone.

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

A minimal configuration might be as follows. In this configuration, the Engineering zone is secured, but all other zones are publicly accessible.

```
appletalk routing
access-list 603 deny zone Engineering
access-list 603 permit additional-zones
access-list 603 permit other-access

interface ethernet 0
appletalk network 3
appletalk distribute-list 603 out
appletalk access-group 603
```

A more comprehensive configuration might be the following, in which the Corporate and MIS zones are public and all other zones are secured:

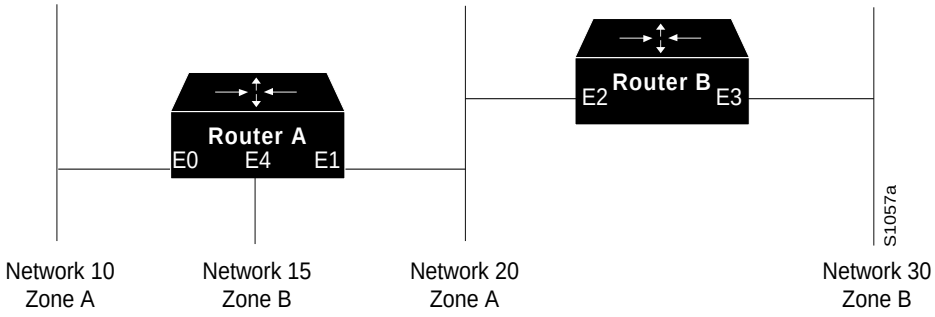
```
appletalk routing
access-list 603 permit zone Corporate
access-list 603 permit zone MIS
access-list 603 deny additional-zones
access-list 603 deny other-access

interface ethernet 0
appletalk network 3
appletalk distribute-list 603 out
appletalk access 603
```

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

Configuring Partial Zone Advertisement

Figure 17-8 illustrates a configuration in which you might want to allow partial advertisement of a particular zone.

Figure 17-8 Example Topology of Partially Obscured Zone



Assume that Router (CommServer) B includes a router-update filter (applied with the **appletalk distribute-list** interface configuration command) on the Ethernet interface 3 that does not accept routing table updates from network 10, nor does it send routing table updates to that network:

```
access-list 612 deny network 10
access-list 612 permit other-access
interface ethernet 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** global configuration command has the following effects:

- If permit-partial-zones is enabled (**appletalk permit-partial-zones**), the routing updates exclude network 10, but include 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, but still include network 15. This is generally considered the preferred behavior and is the default.

Table 17-2 summarizes the associations between the networks shown in Figure 17-8. Table 17-3 details the effects of enabling and disabling partial-zone advertisement with the **appletalk permit-partial-zones** global configuration command.

Table 17-2 Zone and Interface Associations for Partial Zone Advertisement Example

	Network 10	Network 15	Network 20	Network 30
Zone	A	B	A	B
Interfaces	Ethernet 0	Ethernet 4	Ethernet 1 Ethernet 2	Ethernet 3

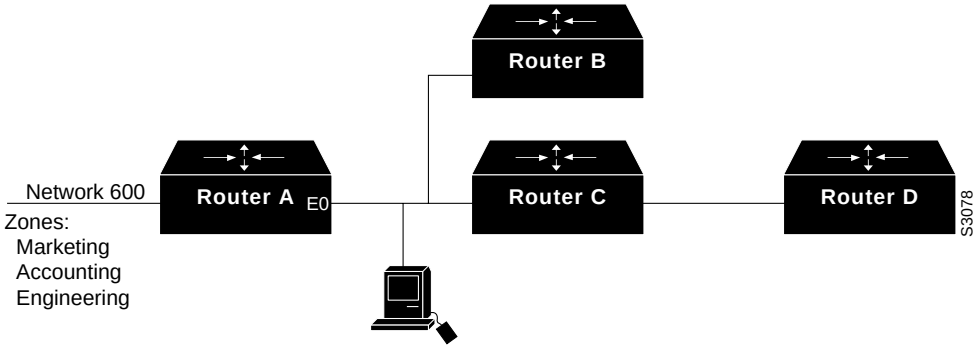
Table 17-3 Partial Zone Advertisement Control on Network 30

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	—

GZL and ZIP Reply Filter Examples

The examples in this section show how to configure GZL and ZIP reply filters, and they illustrate the differences between these two types of filters. Both examples use the configuration shown in Figure 17-9.

Figure 17-9 GZL and ZIP Reply Filters Sample Topology



Both GZL and ZIP reply filters control the zones that can be seen on a network segment. GZL filters control which zones can be seen by Macintoshes on local network segments. These filters have no effect on adjacent routers or communication servers. In order for GZL filters to work properly, all routers or communcation servers on the local segment must be configured with the same access list.

ZIP reply filters control which zones can be seen by adjacent routers and communication servers and by all routers or communication servers downstream from adjacent routers. You can use these filters to hide zones from all Macintoshes on all networks on adjacent routers or communication servers and from all their downstream routers.

Using the configuration shown in Figure 17-9, you would use a GZL filter to prevent the Macintosh on the Ethernet 0 network segment from viewing the zones Engineering and Accounting on network 600. These zones would not be visible via the Macintosh's Chooser. To do this, you configure Router (Communication Server) A as follows:

```
access-list 650 deny zone Engineering
access-list 650 deny zone Accounting
access-list 650 permit additional-zones
!
interface ethernet 0
appletalk getzonelist-filter 650
```

Again using the configuration shown in Figure 17-9, you would use a ZIP reply filter to hide the Engineering and Accounting zones from Routers (Communication Servers) B and C. This filter would also hide the zones from Router (Communication Server) D, which is downstream from Router (Communication Server) C. The effect of this filter is that when these routers request the names of zones on network 600, the zones names Engineering and Accounting will not be returned.

```
access-list 650 deny zone Engineering
access-list 650 deny zone Accounting
access-list 650 permit additional-zones
!
interface ethernet 0
appletalk zip-reply-filter 650
```

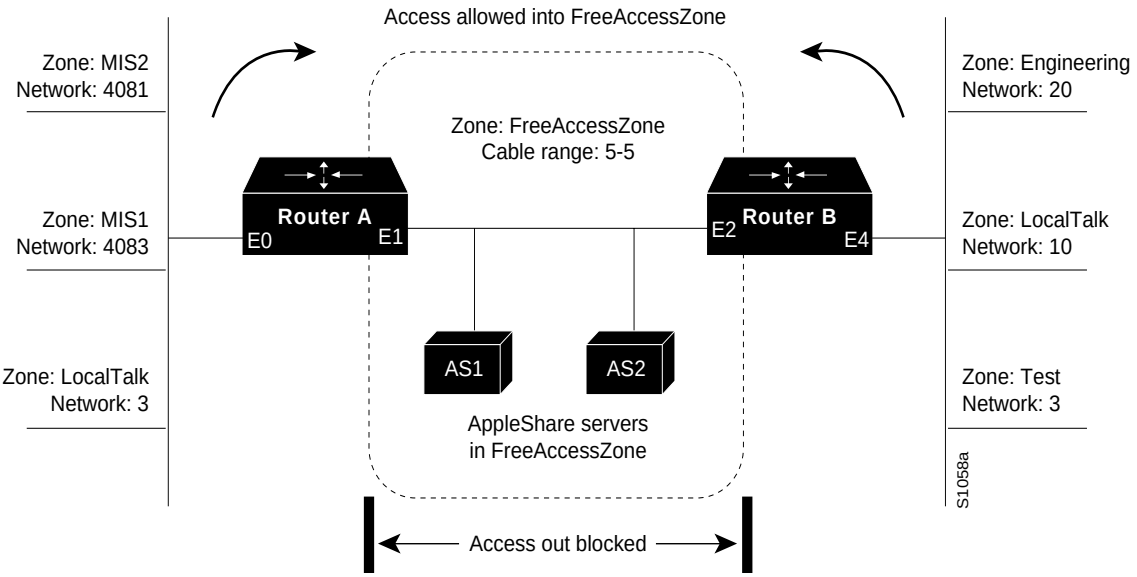
Hiding and Sharing Resources with Access List Examples

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

Establishing a Free-Trade Zone Example

The goal of the configuration shown in Figure 17-10 is to allow all users on all the networks connected to routers (Communication Servers) A and B to be able to access the AppleShare servers AS1 and AS2 in the zone FreeAccessZone. A second requirement is to block cross access through this zone. In other words, users in the zones MIS1, MIS2, and LocalTalk (which are connected to Ethernet interface 0 on Router (Communication Server) A) are not allowed access to any of the resources on networks connected to Ethernet interface 4 on Router (Communication Server) B. Similarly, users in the zones Engineering, Test, and LocalTalk (which are connected to Ethernet interface 4 on Router (Communication Server) B, interface E4) are not allowed access to any of the resources on networks connected to Ethernet interface 0 on Router (Communication Server) A.

Figure 17-10 Controlling Access to Common AppleTalk Network



Note Although there are networks that share the same number on interfaces E0 and E4 and there are 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 (Router A) and E4 (Router B); thus, resources in FreeAccessZone are made accessible to users on all those networks.

The following examples configure Router (Communication Server) A and Router (Communication Server) B for access control illustrated in Figure 17-10. You have to configure only Ethernet interface 1 on Router (Communication Server) A and Ethernet interface 2 on Router (Communication Server) B to provide the desired access.

Configuration for Router (Communication Server) A

```
appletalk routing
!
interface ethernet 1
appletalk cable-range 5-5
appletalk zone FreeAccessZone
appletalk free-trade-zone
```

Configuration for Router (Communication Server) B

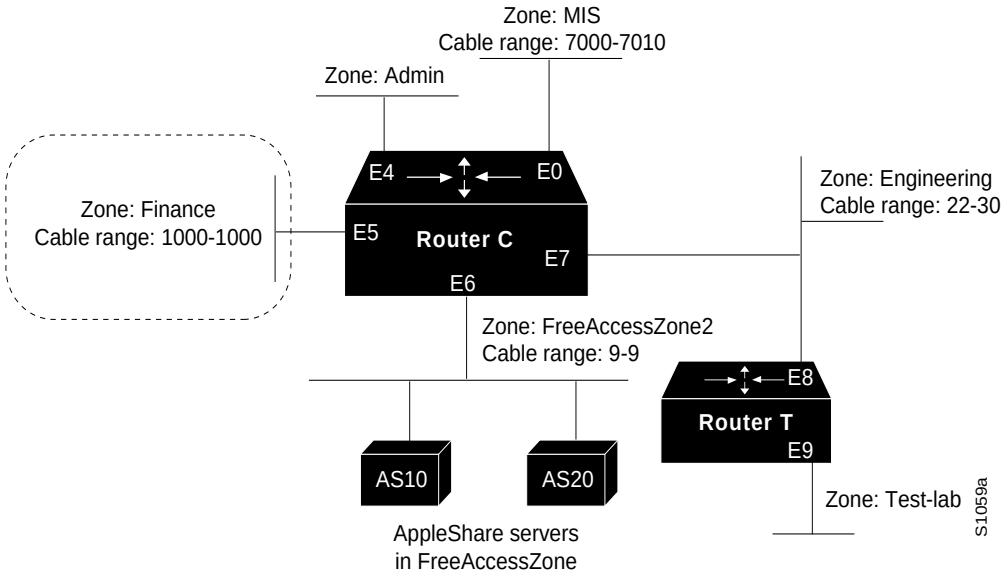
```
appletalk routing
!
interface ethernet 2
appletalk cable-range 5-5
appletalk zone FreeAccessZone
appletalk free-trade-zone
```

When configuring both communication servers, you do not need to define any access lists to prevent users on networks connected to Router (Communication Server) A from accessing resources on networks connected to Router (Communication Server) B, and vice versa. The **appletalk free-trade-zone** interface configuration command implements the necessary restrictions.

Restricting Resource Availability

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 might 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 are denied access to other specific zones. Figure 17-11 illustrates such a situation.

Figure 17-11 Controlling Resource Access among Multiple AppleTalk Zones



The following are the objectives of the configuration in Figure 17-11:

- 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.

To meet these specifications, you define the following access lists:

```
access-list 609 permit cable 9-9
access-list 609 deny other-access
!
access-list 610 permit zone Finance
access-list 610 permit zone FreeAccessZone2
access-list 610 deny additional-zones
!
access-list 611 deny cable-range 1000-1000
access-list 611 deny cable-range 9-9
access-list 611 permit cable-range 7000-7010
access-list 611 permit cable-range 22-30
```

The effects of these access lists are as follows:

- Access list 609 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 611 is intended to be used to accommodate the requirement to allow users in zones Engineering and MIS to mutually access network resources.

Configuration for Ethernet Interface 0

Ethernet interface 0 is associated with the MIS zone. Use the following commands to configure this interface:

```
interface ethernet 0
appletalk cable-range 7000-7010
appletalk zone MIS
appletalk distribute-list 611 out
appletalk distribute-list 611 in
```

Specifying access list 611 results in the following filtering:

- Advertisements of Finance are blocked.
- Advertisements between Engineering and MIS are allowed.

Configuration for Ethernet Interface 5

Ethernet interface 5 is associated with the Finance zone. Use the following commands to configure this interface:

```
interface ethernet 5
appletalk cable-range 1000-1000
appletalk zone Finance
appletalk distribute-list 610 out
appletalk access-group 610
```

The effects of these access lists are as follows:

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

Configuration for Ethernet Interface 6

Ethernet interface 6 is associated with the FreeAccessZone2 zone. Use the following commands to configure this interface:

```
interface ethernet 6
appletalk cable 9-9
appletalk zone FreeAccessZone2
appletalk distribute-list 609 out
appletalk distribute-list 609 in
```

Configuration for Ethernet Interface 7

Ethernet interface 7 is associated with the Engineering zone. The configuration for this interface mirrors that for Ethernet interface 0, because the users in both the MIS and Engineering zones need to have access to each other's resources. Use the following commands to configure Ethernet interface 7:

```
interface ethernet 7
appletalk cable-range 22-30
appletalk zone Engineering
appletalk distribute-list 611 out
appletalk distribute-list 611 in
```

Implicit Configuration of the Admin and Test-Lab Zones

Omitted from the configuration example are any specific configuration commands pertaining to the zones Test-Lab (Ethernet interface 9 on Router (Communication Server) T) and Admin (Ethernet interface 4 on Router (Communication Server) C). No configuration is done for these zones because there are no requirements relating to them listed in the original objectives. The following access control is implicitly handled with the assignment of the stated access lists:

- Users in the Admin zone 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 (Communication Server) T, users in Test-Lab can access the resources in the FreeAccessZone2 and Engineering zones. With the exception of Engineering, no other zones can access resources in Test-Lab.

MacIP Examples

The following example illustrates MacIP support for dynamically addressed MacIP clients with dynamically allocated IP addresses in the range 131.108.0.2 to 131.108.0.10:

```
!Specify server address and zone
appletalk macip server 131.108.0.1 zone Snark
!
!Specify dynamically addressed clients
appletalk macip dynamic 131.108.0.2 131.108.0.10 zone Snark
!
!Assign the address and subnet mask for Ethernet interface 0
interface ethernet 0
ip address 131.108.0.2 255.255.255.0
!
!Enable AppleTalk routing
appletalk routing
!
interface ethernet 0
appletalk cable range 69-69 69.128
appletalk zone Snark
```

The following example illustrates MacIP support for MacIP clients with statically allocated IP addresses:

```
!Specify the server address and zone
appletalk macip server 131.108.0.1 zone Snark
!
!Specify statically addressed clients
appletalk macip static 131.108.0.11 131.108.0.20 zone Snark
appletalk macip static 131.108.0.31 zone Snark
appletalk macip static 131.108.0.41 zone Snark
appletalk macip static 131.108.0.49 zone Snark
!
!Assign the address and subnet mask for Ethernet interface 0
interface ethernet 0
ip address 131.108.0.1 255.255.255.0
!
!Enable AppleTalk routing
appletalk routing
!
interface ethernet 0
appletalk cable range 69-69 69.128
appletalk zone Snark
```

SNMP Example

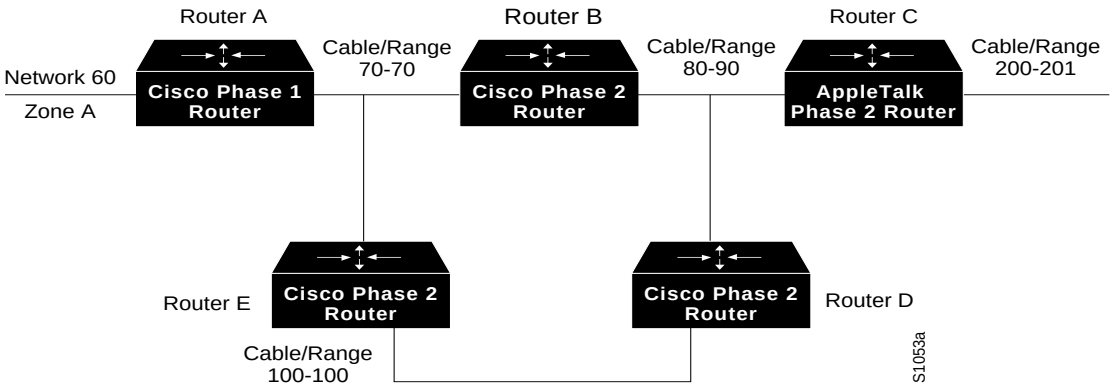
The following example configuration sequence illustrates proper activation of SNMP and AppleTalk on a communication server:

```
!Disable SNMP on the router.
no snmp-server
!
!Enable AppleTalk routing and event logging on the router.
appletalk routing
appletalk event-logging
!
!Configure IP and AppleTalk on Ethernet interface 0.
interface Ethernet 0
ip address 131.108.29.291 255.255.255.0
appletalk cable-range 29-29 29.180
appletalk zone Zombie
!
!Enable SNMP on the router.
snmp-server community propellerhead RW
snmp-server trap-authentication
snmp server host 131.108.2.160 propellerhead
```

Proxy Network Number Example

Assume that your network topology looks like the one in Figure 17-10. Also assume that Router (Communication Server) A supports only nonextended AppleTalk, that Router (Communication Server) B supports only extended AppleTalk (not in transition mode), and that Router (Communication Server) C supports only extended AppleTalk.

Figure 17-12 Example Network Topology



If Router (CommServer) C generates an NBP hookup request for Zone A, Router (Communication Server) B will convert this request to a forward request and send it to Router (Communication Server) A. Since Router (Communication Server) A supports only nonextended AppleTalk, it does not handle the forward request and ignores it. Hence, the NBP lookup from Router (Communication Server) C fails.

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

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

```
appletalk proxy 60 A
```

AppleTalk Enhanced IGRP Example

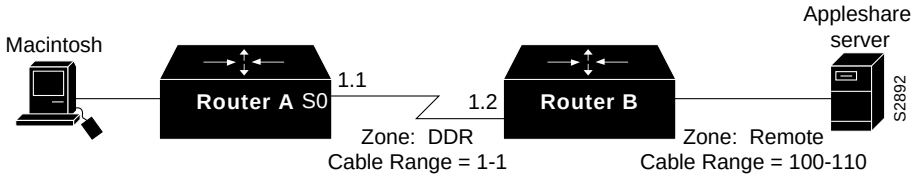
The following example shows how to configure AppleTalk Enhanced IGRP. In this example, Ethernet interface 0 is configured for both Enhanced IGRP and RTMP routing, and serial interface 0 is configured for only AppleTalk Enhanced IGRP routing:

```
appletalk routing eigrp 1
appletalk route-redistribution
!
interface ethernet 0
appletalk cable-range 10-10 10.51
appletalk zone Ethernet 0
appletalk protocol eigrp
!
interface serial 0
appletalk cable-range 111-111 111.51
appletalk zone Serial 0
appletalk protocol eigrp
no appletalk protocol rtmp
```

AppleTalk over DDR Example

The following example describes how to configure AppleTalk to run over a DDR interface, as illustrated in Figure 17-13. When configuring AppleTalk over DDR, you must specify DDR on the interface on which the static neighbor resides before you specify the static route itself. Also, the communication server must know the network address of the static neighbor before you specify the static route. Otherwise, the communication server will not know which interface the static neighbor is connected to. To open an AppleTalk DDR link, there must be at least one AppleTalk access list bound to a dialer group.

Figure 17-13 AppleTalk over DDR Configuration



To configure AppleTalk over DDR, perform the following tasks on Router (Communication Server) A:

Step 1 Configure an access list and dialer group:

```
access-list 601 permit cable 100-110
dialer-list 4 list 601
```

Step 2 Configure the serial interface:

```
interface serial 0
```

```
dialer in-band
dialer string 1234
appletalk cable 1-1 1.1
appletalk zone DDR
dialer-group 4
apple distribute-list 601 in
```

Step 3 Create the static route:

```
appletalk static cable 100-110 to 1.2 zone Remote
```

Step 4 Open the Chooser on the Macintosh.

Step 5 Select any AppleTalk service (like AppleShare, LaserWriter, and so on) in zone Remote. This causes Router (Communication Server) A to dial up Router (Communication Server) B to open a DDR link between them.

Step 6 Select an AppleTalk file server in the zone Remote. After some time, AppleTalk services appear in zone Remote. Select the one that you need.

Step 7 Close the Chooser.

Step 8 Open the AppleTalk session to the remote service.

Step 9 After the AppleTalk session is done, close the connection to the remote service. The DDR link should go down after the DDR idle time has elapsed.

IPTalk Example

This section describes how to set up UNIX-based systems and our routers to use CAP IPTalk and other IPTalk implementations.

The following procedure outlines the basic steps for setting up our routers or communication servers and UNIX hosts for operation using IPTalk implementations.

Note This procedure does not provide full instructions about how to install CAP on the UNIX system. However, it does address the requirements 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 communication server's address and configuration information. Refer to your UNIX system and CAP software manuals for information about building the CAP software and setting up the UNIX startup scripts.

Step 1 Enable AppleTalk routing on all the communication servers that are going to use IPTalk and any communication servers between these routers or communication servers.

Step 2 Enable IP routing on the interfaces that will communicate with the UNIX system. (Refer to the "Configuring IP" and "Configuring IP Routing Protocols" chapters in this publication for more information about configuring IP). These interfaces must be on *the same subnet* as the UNIX system. Also, ensure that IP is enabled on the UNIX system.

Step 3 Allocate an AppleTalk network number for IPTalk. You need a separate AppleTalk network number for each IP subnet that is to run IPTalk.

You can have a number of UNIX machines on the same subnet. They all use the same AppleTalk network number for IPTalk. However, they must have their own individual node identifiers.

It is possible for the same communication server to have IPTalk enabled on several interfaces. Each interface must have a different AppleTalk network number allocated to IPTalk, because 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 older AppleTalk convention that expresses AppleTalk network numbers as two octets (decimal numbers from 0 to 255) separated by a dot. The current AppleTalk convention uses decimal numbers from 1 to 65,279. Use the following formula to convert between the two:

CAP format: *x.y*
Apple format: *d*

- To convert from AppleTalk to CAP:
  $x = d/256$ (/ represents truncating integer division)
  $y = d\%256$ (% represents the remainder of the division)
- To convert from CAP to AppleTalk: $d = x * 256 + y$

Example
AppleTalk format: 14087
CAP format: 55.7

Step 5 Choose a zone name for IPTalk. There are no special constraints on zone name choices. You can use the same zone name for several networks, and you can combine IPTalk and normal AppleTalk networks in the same zone.

Step 6 Decide which UDP ports 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 AppleTalk ports, and their numbers are hardcoded into older versions of CAP. The only problem with using them is that they are not officially assigned by the Internet's Network Information Center (NIC). NIC has assigned a set of UDP ports beginning with 200. Thus, other applications could use them, possibly causing conflicts—although this is unlikely. With CAP releases 5.0 and later, you can configure CAP to use the officially allocated ports. If you do so, RTMP will use port 201, NBP port 202, and so on. Whichever ports you use, you must configure both CAP and the communication server to use the same ones.

Step 7 Enable IPTalk on each interface of the communication server as required. This is illustrated by the following example:

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

In this example, AppleTalk routing is enabled on the interface in two ways:

- Via EtherTalk phase 2, using the cable range 1792–1792 and the zone MIS-Development
- Via IPTalk, using the network number 14087 and the zone MIS-UNIX

Note The node identifier is not specified (that is, it is left as 0) in the **appletalk iptalk** command. The IPTalk node identifier is then 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 identifier would be 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), include the following global configuration command in your configuration:

```
appletalk iptalk-baseport 200
```

Step 8 Configure each UNIX host with a network number, zone name, and router or communication server.

As an example, the following are the contents of the */etc/atalk.local* file from a UNIX system with the IP address 128.6.7.26 and a network mask of 255.255.255.0:

```
# 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, and the second noncomment line defines the address of the router or communication server. In both cases, the first column is 55.7, which is the AppleTalk network number, in CAP format, for use by IPTalk. The second column is the AppleTalk node identifier, which must be the same as the IP host number. The third column on the first line is the zone name, and on the second line it is the IP address of the router or communication server.

Note the following about the entries in the */etc/atalk.local* file:

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

Step 9 Ensure that your CAP software is using the same UDP port numbers as the communication server. Currently, the CAP default is the same as the communication server default, which is port numbers beginning with 768. If you want to use this default, you do not need to take any further action. However, if you want to use the official UDP port numbers (port numbers beginning with 200), ensure that you have included the following command in your communication server configuration:

```
appletalk iptalk-baseport 200
```

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

```
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), previously known 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 to modify */etc/services*.

Configure AppleTalk Control Protocol for PPP Example

The following example illustrates the steps required to set up your access server to accept AppleTalk client requests on interfaces 1 and 3, using the virtual network number 3 and the AppleTalk zone Twiddledee:

```
Cs>enable
Cs#config terminal
CS(config)#appletalk virtual-net 3 Twiddledee
CS(config)#interface async 1
CS(config-int)#encapsulation ppp
CS(config-int)#appletalk client-mode
CS(config-int)#interface async 3
CS(config-int)#encapsulation ppp
CS(config-int)#appletalk client-mode
```