CHAPTER 1

Configuring an AppleTalk Remote Access Server

This chapter describes how to configure your communication server to act as an AppleTalk Remote Access (ARA) server. It does not describe how to configure or use the client Macintosh. Refer to Apple Computer's *Apple Remote Access Client User's Guide* and the *Apple Remote Access Personal Server User's Guide* for information about how to use ARA software on your Macintosh. For a complete description of the commands in this chapter, refer to Chapter 2.

Cisco's Implementation of ARA

Cisco's implementation of ARA gives Macintosh users direct access to information and resources in remote locations. Macintosh users can connect to another Macintosh computer or AppleTalk network over standard telephone lines. For example, if you have a PowerBook at home and need to get a file from your Macintosh at the office, ARA software can make the connection between your home and office computers.

You can configure your communication server to act as an ARA server by enabling AppleTalk and ARA Protocol. Configuring your communication server to act as an ARA server allows remote Macintosh users to dial in, become a network node, and connect to devices on other networks. ARA Protocol support on the communication server is transparent to the Macintosh end user.

The following Macintosh and communication server software support is required for ARA connectivity:

- A Macintosh running ARA software and a connection control language (CCL) script.
- A communication server configured as an ARA server.

Figure 1-1 shows how your communication server can act as an ARA server between remote Macintosh computers (in Figure 1-1, a Macintosh SE and a PowerBook) and devices on another network.

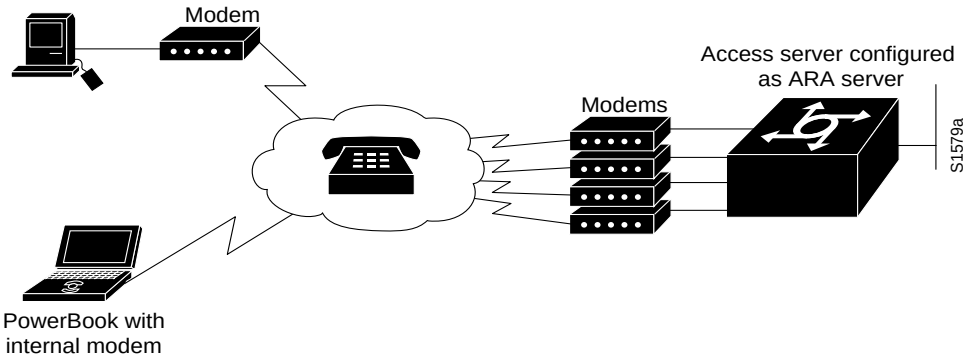

Figure 1-1 ARA Configuration Overview

ARA Protocol

Enabling ARA on your communication server permits the server to support ARA on the Macintosh and, therefore, to act as an ARA server.

AppleTalk

AppleTalk is a client-server, or distributed, protocol. AppleTalk users share network resources, such as files and printers, with other users. Interactions with different servers are transparent to users, because the computer determines the location of the requested material and accesses it without requesting information from the user.

AppleTalk identifies several network entities: node, network, and zone. A node is any device connected to an AppleTalk network. The most common nodes are Macintosh computers and laser printers, but many other types of computers are also capable of AppleTalk communication, including IBM PCs, Digital VAX/VMS systems, and a variety of workstations. A communication server, which provides only one network interface, is considered a node on the network. In this chapter, the term *router* refers to any device that routes AppleTalk packets. An AppleTalk network is a single logical cable, and an AppleTalk zone is a logical group of one or more (possibly noncontiguous) networks.

Apple Computer has produced a variety of internetworking products with which to connect AppleTalk local-area networks. Apple supports Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), and its own proprietary twisted-pair media access system (called LocalTalk).

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

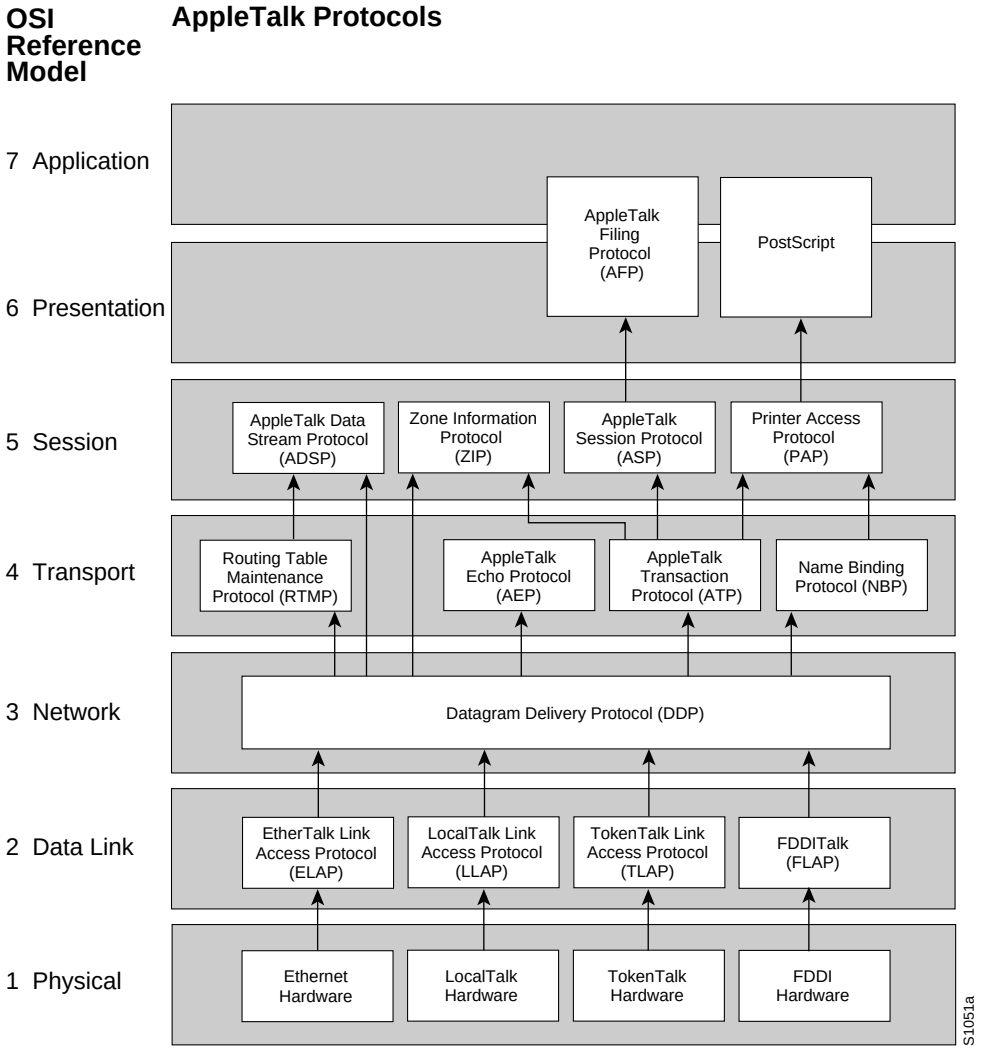

Figure 1-2 AppleTalk and the OSI Reference Model

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

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

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

The Cisco AppleTalk implementation also includes the following enhancements:

- Support for EtherTalk 1.2 and EtherTalk 2.0
- Support for serial protocols, including SMDS, Frame Relay, X.25, and HDLC
- Configurable protocol constants
- No software limits on the number of zones
- MacTCP support via the MacIP server
- NBP proxy service providing compatibility between AppleTalk Phase 1 and AppleTalk Phase 2
- Access control support to allow filtering of zones, routing data, and packets
- Integrated node name support to simplify AppleTalk management
- Interactive access to AEP and NBP provided via the **ping** command
- Support for both configured (called *seed*) and discovered configuration
- Responder support used by Inter•Poll and other network monitoring packages

---

Note Apple Computer uses the name *AppleTalk* to refer the Apple Networking Architecture (ANA), whereas the actual transmission media used to form an AppleTalk network are referred to as LocalTalk (Apple Computer's proprietary twisted-pair transmission medium for AppleTalk), TokenTalk (AppleTalk over Token Ring), and EtherTalk (AppleTalk over Ethernet).

AppleTalk, like many network protocols, makes no provision for network security. The AppleTalk protocol architecture requires that security measures be executed at higher application levels. The communication server software supports AppleTalk network access lists, providing filters at the packet level.

Extended (Phase 2) versus Nonextended (Phase 1) AppleTalk

AppleTalk was designed for local work groups. With the installation of over 1.5 million Macintosh computers in the first five years of the product's life, Apple found that some large corporations were exceeding the design limits of AppleTalk. Apple's solution was to create extended AppleTalk. The extended AppleTalk architecture increases the number of nodes per AppleTalk internetwork to over 16 million and an unlimited number of zones per cable.

The introduction of the extended AppleTalk architecture also introduced the concept of nonextended and extended networks. Nonextended AppleTalk networks are sometimes called "Phase 1," and extended networks are called "Phase 2." Nonextended networks refer to the nonextended AppleTalk Ethernet 1.0 networks (no longer supported by Apple but still supported by Cisco), and to the nonextended serial line-based networks, including those configured using X.25 and LocalTalk.

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

The AppleTalk extended-network architecture provides extensions compatible with nonextended AppleTalk internetworks. The AppleTalk extended architecture was designed to remove the previous limits of 254 concurrently active AppleTalk nodes per cable, as well as the previous limit of one

AppleTalk zone per cable. Extended AppleTalk contains better algorithms for choosing the best routers for traffic and is designed to minimize the amount of broadcast traffic generated for routing updates.

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

Nonextended AppleTalk Addressing

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

AppleTalk Zones

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

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

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

A zone can include many networks that need not be located together physically. A zone is not limited by geographical area. The partitioning afforded by zone names is conceptual, not physical.

The network manager defines zones when he or she configures a router. For nonextended networks, each AppleTalk-configured interface must be associated with exactly one zone name, and for extended networks, each AppleTalk-configured interface can be associated with one or more zone names. Until a zone name has been assigned, AppleTalk capability is disabled for that interface. The section "Configure AppleTalk" later in this chapter lists the commands to use in the zone-naming process. Refer to Chapter 2 for a description of each command and its guidelines for usage.

Name Binding Protocol

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

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

- Name registration
- Name deletion
- Name lookup
- Name confirmation

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

Zone Information Protocol

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

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

Dynamic Configuration

AppleTalk supports dynamic configuration (discovery mode). Not all fields of an AppleTalk address need to be specified to configure an AppleTalk router. If there is another AppleTalk communication server or router on the network, it might be able to supply the network number and zone name. A preconfigured router on an AppleTalk network acts as a seed router or communication server, responding to configuration queries from other nodes on its network.

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

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

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

Table 1-1 Examples of AppleTalk Addresses

AppleTalk Address	Description
34.5	Completely specified (network 34, node 5)
0.5	Partially qualified (network unspecified, node 5)
122.0	Partially qualified (network 122, node unspecified)
0.0	Completely unspecified

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

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

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

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

Extended AppleTalk Addressing

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

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

A new concept of cable ranges is introduced with the extended AppleTalk. Ranges of network numbers and multiple zones can exist on a single logical cable. But the node can exist in only one zone and on only one network.

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

AppleTalk Name Registration

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

```
mycommserver.Ethernet0:ciscoRouter@Engineering
```

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

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

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

```
Net  Adr Skt Name                    Type            Zone
4042   8 254 brown.Ethernet0         ciscoRouter  Engineering
4028   8 254 brown.Async1            ciscoRouter  Engineering
```

AppleTalk Responder Support

The communication server answers AppleTalk responder requests. The listener is installed on the AppleTalk interface name registration socket.

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

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

The information is returned as follows:

- System bootstrap version (ROM version)
- Currently running software version
- AppleTalk version—this always indicates 56, which is the first Apple Macintosh version that contained AppleTalk Phase 2 support
- AppleTalk responder version—this always displays 100, which indicates support of Version 1.0 responder packets
- Report that AppleShare is not installed

Figure 1-3 illustrates a typical output display for Inter•Poll that lists this information.

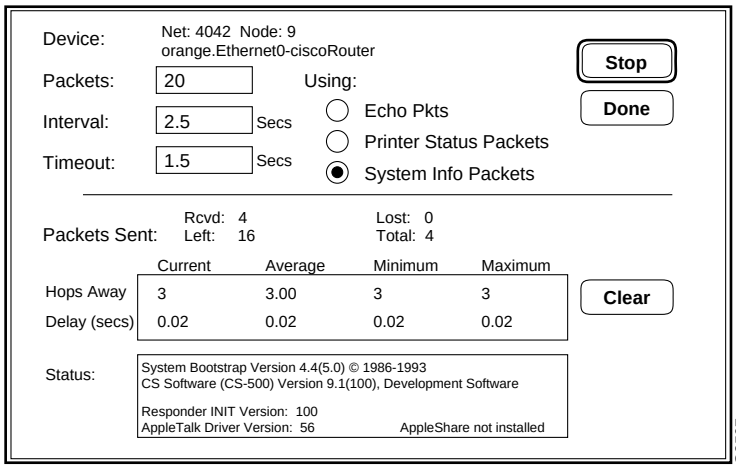

Figure 1-3 Inter•Poll Output

ARA Task Overview

To set up your communication server as an ARA server, complete the following tasks:

- Connect cables (page 1-9)
- Configure the line and the modem (page 1-11)
- Configure AppleTalk (page 1-12)

The following tasks are optional:

- Customize ARA (page 1-15)
- Customize the AppleTalk configuration (page 1-17)
- Configure system security (page 1-22)
- Monitor and debug an ARA server (page 1-24)

Connect Cables

Figure 1-4 shows how to connect a Macintosh directly to the communication server and how to connect a Macintosh by means of internal and external modems. The directly connected Macintosh can be used as a terminal from which you can configure the communication server.

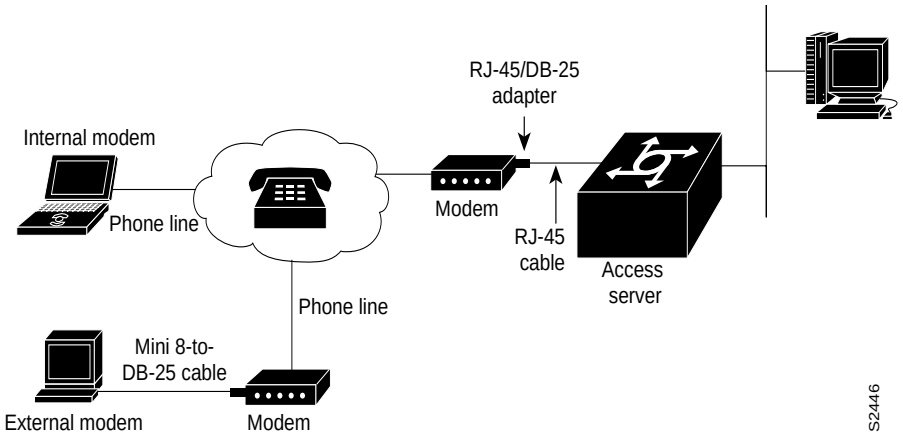

Figure 1-4 ARA Server Cabling and Connections

To connect a Macintosh directly to the communication server, use the FDTE version of the RJ-45-to-DB-25 adapter (Cisco Part Number: 29-FDTE-02) to connect the "rolled" RJ-45 cable from the communication server to the Mini 8-to-DB-25 cable from the Macintosh.

To connect a modem to the communication server, use the MMOD version of the RJ-45-to-DB-25 adapter to connect the "rolled" RJ-45 cable from the communication server to the modem. You can also use a Cisco MDCE adapter that you have modified by moving the DB-25 pin in position 6 to position 8.

Figure 1-5 shows the pins of a Mini 8 connector, the pins of a DB-25 connector, and how they are connected.

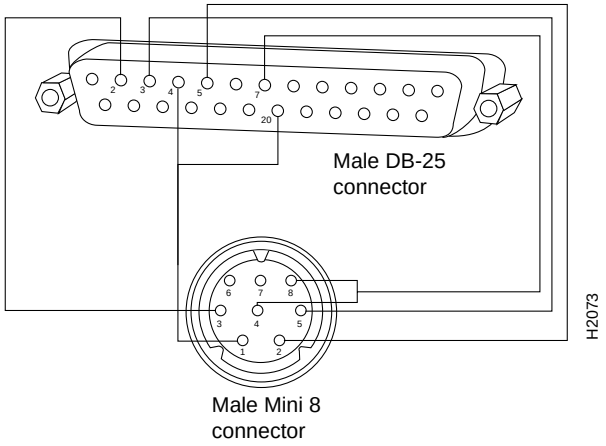

Figure 1-5 Wiring Diagram of a Mini 8-to-DB-25 Cable

Table 1-2 explains the pin functions.

Table 1-2 Building a Mini 8-to-DB-25 Cable

Din-8 Pin Number	Din-8 Pin Function	DB-25 Pin Number	DB-25 Pin Function
1	Output handshake	4, 20	RTS, DTR
2	Input handshake/external clock	5	CTS
3	TxD	2	TxD
4, 8	Ground, RxD(-)	7	Ground
5	RxD(+)	3	RxD

Note This cable implements hardware flow control. It allows the Macintosh to assert both the DTR and the RTS signals with the HSK0 control line. The HSK1 control line is attached to pin 5, which allows the Macintosh to monitor the CTS signal from the modem. Data is transmitted to the modem on pin 2 of the DB-25 connector and received from the modem on pin 3 of the DB-25 connector. Pin 7 on the DB-25 connector grounds the connection between the Macintosh and the modem. Because DTR is tied to RTS, you should configure the modem to ignore any change in the state of DTR. Otherwise, an RTS flow control change would cause the modem to hang up the telephone line. For more information about cables, connectors, and adapters, see the hardware installation and maintenance manual for your communication server.

Configure the Line and the Modem

Configure the line on the communication server as follows:

- Specify line speed—38400 bps on high-speed modems is recommended
- Set hardware flow control—use the **flowcontrol hardware** command to enable hardware flow control
- Specify your dial-in type—use the **modem inout** command to configure the line for both incoming and outgoing calls, or use the **modem ri-is-cd** command to configure the line for incoming calls only

Note The **autobaud** command is not supported with ARA.

Configure the modem as follows:

- Set hardware flow control
- Disable software flow control (XON/XOFF)
- Disable echo
- Set quiet mode (that is, prevent the modem from responding to commands)
- Set auto-answer to answer on 1 ring (2 rings are required in Germany)
- Set modem so that DSR follows CD
- Reset to nonvolatile random-access memory (NVRAM) when DTR drops

If your modem does not support this configuration, see the *Communication Server Configuration Guide* or the *Communication Server Command Reference* publication for information about configuring a line to support your modem.

Configure AppleTalk

To configure ARA on your communication server, you need to perform the following tasks:

- Enable AppleTalk
- Configure an AppleTalk interface
- Enable ARA

The sections that follow describe each of these tasks. See Chapter 2 for information about commands listed in these tasks.

Enable AppleTalk Service

To enable the AppleTalk service in global configuration mode, perform the following task:

Task	Command
Enable AppleTalk.	**appletalk service**

Configure an AppleTalk Interface

You can manually configure an interface for AppleTalk or, if an interface is connected to a network that has at least one other communication server or router configured for AppleTalk, you can dynamically configure the interface using discovery mode.

If the internet already exists, the zone and cable range must match the existing configuration. To identify existing cable ranges and zone names, configure the communication server for discovery mode.

You can also configure an AppleTalk interface on a segment for which there are no AppleTalk routers.

Manual Interface Configuration

To manually configure an interface for extended AppleTalk, perform the following tasks:

Task	Command
Specify an interface.	**interface** *type unit*
Assign a cable range to an interface.	**appletalk cable-range** *cable-range* [*network.node*]
Assign a zone name to the interface.	**appletalk zone** *zone-name*

If you assign more than one zone name, the first name you assign is the default zone.

You can define up to 255 unique zone names.

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

Dynamic Interface Configuration

If an AppleTalk 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 communication server or router and then uses this information to configure itself. Once the interface has been configured, you can manually enter the dynamically acquired information.

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

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

Note Discovery mode does not work with synchronous serial lines.

If there is no operational communication server or router on the attached network, you must manually configure the interface as described in the earlier section "Manual Interface Configuration." Also, if a discovery-mode interface is restarted, another operational communication server or router must be present before the interface can become operational.

A communication server starts up by first acquiring its configuration from memory. Then, if an interface is not configured for discovery mode, the interface starts up as follows:

- The interface must be configured with the **appletalk address** or **appletalk cable-range** command and the **appletalk zone** command.
- If the interface is properly configured, the interface attempts to verify the stored configuration with another communication server or router on the attached network.
- If there is any discrepancy, the interface does not start up.
- If there are no neighboring operational communication servers or routers, the communication server assumes the stored configuration is correct, and the interface becomes operational.

Using discovery mode does not affect an interface's ability to respond to configuration queries from other 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 acquires the zone name from another interface.

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

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. These methods are described in the sections that follow.

Method 1

In the first method, you immediately put 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 configure an interface for discovery mode using this method, perform the following tasks:

Task	Command
Specify an interface.	**interface** *type unit*
Put the interface into discovery mode by assigning it the cable range 0-0.	**appletalk cable-range 0–0**

Method 2

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. To configure an interface for discovery mode using this method, perform the following tasks:

Task	Command
Specify an interface.	**interface** *type unit*
Assign an AppleTalk address to the interface.	**appletalk cable-range** *cable-range* [*network.node*]
Put the interface into discovery mode.	**appletalk discovery**

Configuring a Segment That Has No Routers

You can also configure an AppleTalk interface on a LAN segment that does not have any AppleTalk routers by performing the following tasks:

Task	Command
Turn on AppleTalk, but do not enable routing.	**appletalk service**
Specify an interface.	**interface e 0**
Specify the AppleTalk address as 1, which is the default address when there are no routers.	**appletalk address 1.1**
Specify the name of the local zone.	**appletalk zone ***

Note that you cannot use discovery mode for this configuration.

Enable ARA

To enable ARA on a line, perform the following tasks:

Task	Command
Specify a line or lines.	**line** {*number* | [*start-number end-number*]}
Enable ARA on a line.	**arap enable**

Customize ARA

The commands in this section can be used to customize ARA support. Some of the commands are required for certain configurations. Possible functions include the following:

- Configure automatic protocol startup (page 1-15)
- Set a dedicated ARA line (page 1-16)
- Set the session time limit (page 1-16)
- Set the disconnect warning time (page 1-16)
- Disallow guests (page 1-16)
- Control access (page 1-16)

Note ARA does not support the **autobaud** command.

The following sections describe these tasks. See Chapter 2 for information about commands listed in these tasks.

Configure Automatic Protocol Startup

To configure the communication server to automatically start an ARA session, perform the following tasks in global configuration mode:

Task	Command
Specify a line in global configuration mode.	**line** {*number* | [*start-number end-number*]}
Configure a line to automatically start an ARA session.	**autoselect**

The **autoselect** command permits the communication server to automatically start an appropriate process when a starting character is received. The communication server detects either a Return character, which is the start character for an EXEC session, or the start character for the ARA protocol.

This command is required for all ARA-enabled lines that are not configured as dedicated ARA lines and that are not configured for TACACS logins.

Note The **autoselect** command should not be used with TACACS.

Set a Dedicated ARA Line

To set a line to function only as an ARA connection, perform the following task in line configuration mode:

Task	Command
Configure a line for ARA only.	**arap dedicated**

Alternatives are to set the line for **autoselect** or TACACS logins.

Set the Session Time Limit

To set the maximum length of an ARA session for a line, perform the following task in line configuration mode:

Task	Command
Set the maximum length of an ARA session.	**arap timelimit** [*minutes*]

The default is to have unlimited length connections. This task is optional.

Set the Disconnect Warning Time

To configure when to display a disconnect warning, perform the following task in line configuration mode:

Task	Command
Set when a disconnect warning message will be displayed, in number of minutes before the line is set to disconnect.	**arap warningtime** [*minutes*]

This command is only valid if a session time limit is set.

Disallow Guests

A guest is a person who connects to the network without having to give a name or a password. To prohibit Macintosh guests from logging in through the communication server, perform the following task in line configuration mode:

Task	Command
Prohibit guests from logging in to the ARA network.	**arap noguest**

Caution Do not enter the **arap noguest** command if TACACS is enabled.

Control Access

You can control Macintosh access to zones and networks by using **arap** commands to reference access control lists configured using AppleTalk **access-list** commands.

To control what zones the Macintosh user will see, perform the following task in line configuration mode:

Task	Command
Limit the zones the Macintosh user sees.	**arap zonelist** *zone-access-list-number*

To control traffic from the Macintosh to networks, perform the following task in line configuration mode:

Task	Command
Control access to networks.	**arap net-access-list** *net-access-list-number*

Customize the AppleTalk Configuration

To customize the AppleTalk configuration, complete the following tasks:

- Disable checksum generation and verification (page 1-17)
- Configure MacIP (page 1-18)
- Control Access to AppleTalk Networks (page 1-20)

This section describes how to perform these configuration tasks. See Chapter 2 for information about commands listed in these tasks.

Figure 1-6 shows a configuration in which a communication server acting as an ARA server is serving a local network that is not connected to an internet.

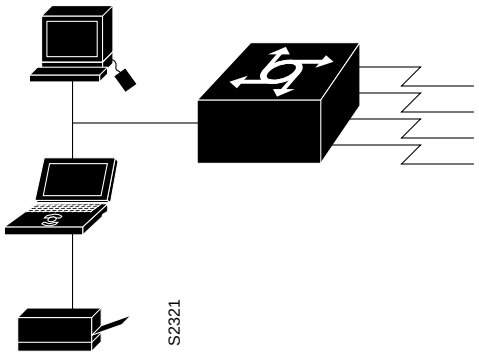

Figure 1-6 ARA Server Not on an Internet

Disable Checksum Generation and Verification

By default, the communication server generates checksums for all ARA traffic that requests them. You might want to disable checksum generation and verification as if you have an older LaserWriter printer or other device that cannot receive packets with checksums.

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

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

Configure MacIP

The communication server implements MacIP, a protocol that routes IP datagrams to IP clients using AppleTalk Datagram Delivery Protocol (DDP) low-level encapsulation. MacIP allows the communication server to assign an ID number to a Macintosh computer that dials in. The ID number allows the Macintosh computer to run MacTCP applications.

Cisco 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*. This implementation of MacIP conforms to the September 1991 draft RFC with the following exceptions:

- Communication servers do not fragment IP datagrams that exceed the DDP MTU and that are bound for DDP clients of MacIP.
- Communication servers do not route to DDP clients outside of configured MacIP client ranges.

MacIP is required to provide access to IP network servers for those users. It is also required for environments in which Macintosh users use ARA or are connected to the network using LocalTalk or PhoneNet cabling systems.

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. 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 is an important disadvantage in implementing MacIP on a communication server: 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 the Cisco communication server, AppleTalk must be configured on the communication server as follows:

- AppleTalk must be enabled.
- IP must be enabled.
- The MacIP zone name you configure must be associated with a configured or seeded zone name.
- If you are using MacIP to allow Macintosh computers to communicate with IP hosts on the same LAN segment (that is, the Macintosh computers are on the Cisco interface on which MacIP is configured) and the IP hosts have extended IP access lists, these access lists should include entries to permit IP traffic destined for these IP hosts (from the MacIP addresses). If these entries are not present, packets destined for IP hosts on the local segment will be blocked (that is, they will not be forwarded).

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 no form of the resource address assignment command (such as the **no appletalk macip dynamic zone** *server-zone* command) 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*

A MacIP server is not registered using 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 “Example of Configuring MacIP.”

Static addresses are for users who require fixed addresses for IP DNS services and for administrators who do not want addresses to change so 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 “Example of Configuring MacIP.”

In general, you should 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.

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 use zone names to regulate access to the internetwork. Zone names are good control points, because they are the only network-level abstraction that users can access. You can express zone 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.

Because AppleTalk-style access lists are based on zones, 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 is useful for defining access lists that control the disposition of networks that overlap, are contained by, or exactly match a specific network number range.

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 and numeric network entries.

There are two types of filters you can use on AppleTalk networks:

- Data packet filters
- GetZoneList (GZL) filters

AppleTalk network access control differs from that of other protocols in that the order of the entries in an access list is unimportant. However, there are still some constraints you need to keep 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 an **access-list permit network** *xxx* statementand then enter an **access-list deny network** *xxx* statement. 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 that do not satisfy any of the access control statements in the access list.

  To explicitly specify how you want these packets to be handled, use the **access-list other-access** command when defining access conditions for networks and cable ranges, and use the **access-list additional-zones** command when defining access conditions for zones. If you use one of these

commands, it does not matter where in the list you put it: the router software automatically puts 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 that do not satisfy any of the access control statements in the access list, the packets 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.

- Create access lists.
- Create filters.

Create Access Lists

An access list defines the conditions used to filter packets sent out of the interface. (These conditions are sometimes also used to filter incoming packets.) 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 only specify one each of the commands that specify default actions to take if none of the access conditions are matched. That is, 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, perform one or more of the following tasks in global configuration mode:

Task	Command
Define access for a single cable range (for extended networks only).	**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** *start-end*
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**

The access list number can be a decimal value from 600 to 699.

To create access lists that define access conditions for zones, 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**

The access list number can be a decimal value from 600 to 699.

Configure System Security

Two types of security can be used on your communication server when it is acting as an ARA server:

- Internal user authentication, with username and password information stored on the communication server
- TACACS user authentication, with username and password information stored on a TACACS server

The following sections describe these tasks. See Chapter 2 for information about the commands listed in these tasks.

Configure Internal Username Authentication

To configure your communication server for internal username authentication, perform the following task in global configuration mode:

Task	Command
Specify a username and password.	**username** *name* **password** *password*

Enter this information for each supported user.

Configure TACACS Security

You can use TACACS security if you have configured a TACACS server and have a CCL script that allows you to use TACACS security. This section tells you how to modify your CCL script so that you can use TACACS security and how to configure a line to use a TACACS server for user authentication.

Modify Scripts to Support TACACS

To use AppleTalk Remote Access with TACACS, you must modify your CCL scripts. For a number of popular modems, we provide CCL files that you can use to modify your CCL scripts to support TACACS security. This section explains how to use the CCL files provided by us to modify AppleTalk Remote Access CCL scripts to work with TACACS security.

We recommend using the ARA Modem Toolkit provided through the AppleTalk Programmers and Developers Association (APDA); it provides both syntax checking and a script player.

AppleTalk Remote Access CCL scripts are primarily used to work with modems to make connections to remote machines. When the connection has been established, the script ends and ARA is activated. TACACS authentication occurs after the connection is established but before the protocol becomes active.

Insert TACACS logic just before the end of a script. The CCL TACACS logic performs the following user authentication tasks:

- When the “Username:” prompt is transmitted from the communication server, the user's name is obtained from the Macintosh and sent to the TACACS server.
- When the “Password:” prompt is transmitted, the user’s password is obtained from the Macintosh and sent to the TACACS server.

- After a successful login, indicated by an EXEC prompt at the communication server, the EXEC command **arap** is sent.

The script ends and ARA begins.

CCL scripts control logical flow by jumping to labels. The labels are the numbers 1 through 128, and will not necessarily be in sequential order in the script file. The TACACS logic in CCL files provided by us have label numbers from 100 through 127. In most environments, copy the complete TACACS logic from an existing file.

The steps for creating a new TACACS CCL file are as follows:

Step 1 Copy the TACACS logic from the CCL file provided by Cisco into the file being modified.

Step 2 Locate the logical end of the script and insert the command **jump 100**.

Copying the TACACS Logic

In most cases, you can simply insert the TACACS logic at the appropriate place in your CCL script. The one case that requires extra attention is when the original CCL script has labels that conflict with the Cisco logic. The labels must be resolved on a case-by-case basis, usually by changing the label numbers used by the original script. This is a fairly simple programming job, but you should read and understand the manual that comes with the Modem Toolkit before beginning.

Locating the Logical End of the Script

You can locate the logical end of the script by following its flow. Most scripts have the following basic structure:

- Initialize the modem
- Dial the number
- After connection, display the connection speed
- Exit

The characteristic logical end of the script is as follows:

```
@label N
! N is any integer between 1 and 128
if ANSWER N+1
! If we're answering the phone, jump directly
! to the label N+1
pause 30
! We're not answering the phone, therefore we
! must be calling. Wait three seconds for the
! modems to sync up.
@label N+1
exit 0
! quit and start up ARA
```

It is common in this case to replace “pause 30” with “jump 100.” In fact, this is usually the only change made to the logic of the original script.

Configure TACACS Server User Authentication

To configure a line to use a TACACS server for user authentication, perform the following tasks:

Task	Command
Specify line or lines.	**line** {*number* | [*start-number end-number*]}
Use a TACACS server for user authentication.	**login tacacs**

Figure 1-7 shows the TACACS login screen on the Macintosh.

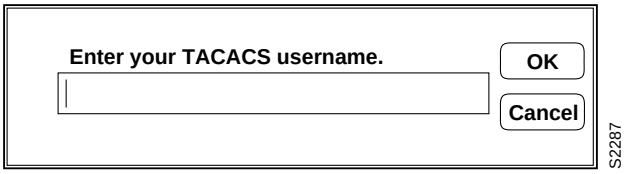

Figure 1-7 TACACS Login Screen on the Macintosh

Figure 1-8 shows the TACACS password screen on the Macintosh.

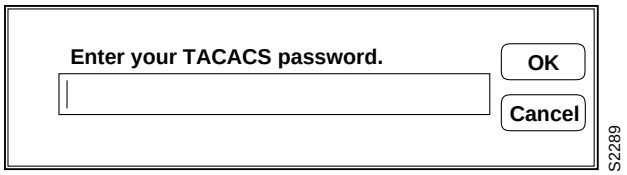

Figure 1-8 TACACS Password Screen on the Macintosh

See the *Communication Server Configuration Guide* or the *Communication Server Command Reference* publication for more information about configuring TACACS security.

Monitor and Debug an ARA Server

To display information about a running ARA connection, perform the following task in privileged EXEC mode (reached by entering the **enable** command and a password):

Task	Command
Display information about a running ARA connection.	**show arap** [*line-number*]

The **show arap** command with no arguments displays a summary of ARA traffic since the communication server was last booted. The **show arap** command with a specified line number displays information about the connection on that line.

Monitor the AppleTalk Network

The communication server software provides several commands you can use to monitor an AppleTalk network. In addition, you can use Apple Computer's Inter•Poll, which is a tool 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 Inter•Poll.

To monitor the AppleTalk network, perform one or more of the following tasks:

Task	Command
List the entries in the AppleTalk ARP table.	**show appletalk arp**
Display AppleTalk-related interface settings.	**show appletalk interface** [**brief**] [*interface unit*]
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 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*]

Debug the ARA Server

To debug ARA connections, perform the following tasks in privileged EXEC mode:

Task	Command
Debug internal ARA packets.	**debug arap internal**
Debug memory allocation for ARA.	**debug arap memory**
Debug low-level asynchronous serial protocol.	**debug arap mnp4**
Debug compression.	**debug arap v42bis**

Configuration Examples

This section contains examples of ARA configuration on the communication server.

Example of Configuring an Extended AppleTalk Network

The following example configures the interface for an extended AppleTalk network. It defines the zones Orange and Brown. The cable range of one allows compatibility with nonextended AppleTalk networks.

```
appletalk service
interface ethernet 0
appletalk cable-range 69-69 69.128
appletalk zone Orange
appletalk zone Brown
```

Example of Configuring an Extended Network in Discovery Mode

The following example configures an extended network in discovery mode. In Figure 1-9, communication server A provides the zone and network number information to the interface when it starts.

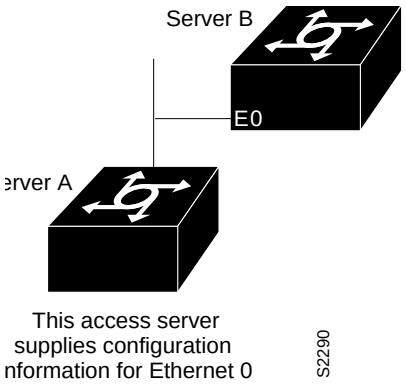

Figure 1-9 Discovery Mode

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

```
appletalk service
interface ethernet 0
appletalk cable-range 0-0 0.0
```

Example of Configuring ARA

The following example configures the communication server for ARA support, as described in the comments (lines beginning with an exclamation point [!]).

```
! Enable AppleTalk on the communication server
appletalk service
!
interface Ethernet 0
ip address 128.66.1.1 255.255.255.0
!
! On interface Ethernet 0, assign network number 103 to the physical cable and
! assign zone name “Marketing Lab” to the interface. Assign a zone name if
! you are creating a new AppleTalk internet. If the internet already exists,
! the zone and cable range must match exactly, or you can leave the cable
! range at 0 to enter discovery mode. The suggested AppleTalk
! address for the interface in this example is 103.1
interface Ethernet 0
appletalk cable-range 103-103 103.1
appletalk zone Marketing Lab
! Configure a username and password for the communication server.
username jake password sesame
! On lines 4 through 8, InOut modems are specified, the lines are configured
! to automatically start an EXEC session or enable AppleTalk, AppleTalk Remote
! Access Protocol is enabled, the modem speed is specified as 38400 bps, and
! hardware flow control is enabled.
line 4 8
modem InOut
autoselect
arap enabled
speed 38400
flowcontrol hardware
```

Note that you must set your terminal emulator to match the speed that you set for the line.

Example of Expanding the Cable Range

In the following example, the cable range is changed and the zone name is reentered.

The initial configuration is as follows:

```
appletalk cable-range 100-103
appletalk zone Twilight Zone
```

The cable range is expanded as follows:

```
appletalk cable-range 100-109
```

At this point, you must reenter the zone name as follows:

```
appletalk zone Twilight Zone
```

Example of Configuring MacIP

The following example illustrates MacIP support for dynamically addressed MacIP clients with dynamically allocated IP addresses in the range 131.108.8.2 to 131.108.8.10:

```
! Specify server address and zone
appletalk macip server 131.108.8.1 zone Snark

! Specify dynamically addressed clients
appletalk macip dynamic 131.108.8.2 131.108.8.10 zone Snark
!
! Assign the address and subnet mask for Ethernet interface 0
interface ethernet 0
ip address 131.108.8.1 255.255.255.0
!
! Enable AppleTalk service
appletalk service
!
interface ethernet 0
appletalk cable range 69-69 69.128
appletalk zone Snark
!
! Specify server address and zone
appletalk macip server 131.108.8.1 zone Snark
!
! Specify dynamically addressed clients
appletalk macip dynamic 131.108.8.2 131.108.8.10 zone Snark
```

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

```
! Assign the address and subnet mask for Ethernet interface 0
interface ethernet 0
ip address 131.108.8.1 255.255.255.0
!
! Enable AppleTalk
appletalk service
!
interface ethernet 0
appletalk cable range 69-69 69.128
appletalk zone Snark
! Specify the server address and zone
appletalk macip server 131.108.8.1 zone Snark
!
Specify statically addressed clients
appletalk macip static 131.108.8.11 131.108.8.20 zone Snark
appletalk macip static 131.108.8.31 zone Snark
appletalk macip static 131.108.8.41 zone Snark
appletalk macip static 131.108.8.49 zone Snark
```

Example of Configuring TACACS Username Authentication

In the following example, line 1 is configured for ARA and username authentication will be performed on a TACACS server:

```
line 1
login tacacs
arap enable
```

Caution Do not use the **autoselect** command if TACACS is enabled.

Example of Configuring a Dedicated ARA Line

In the following example, line 2 is configured as a dedicated ARA line, user authentication information is configured on the ARA server, and guests are disallowed from making ARA sessions:

```
username jsmith password woof
line 2
arap dedicated
arap noguest
```

Example of Configuring a Multiuse Line

In the following configuration, ARA is enabled on lines 2 through 16, username authentication is configured on the ARA server, and the lines are configured to automatically start an ARA session when an ARA user on a Macintosh attempts a connection:

```
username jsmith password woof
line 2 16
autoselect
arap enabled
arap noguest
```

Example of Configuring an ARA Server

The following example shows how to set up ARA functionality on a communication server.

Log in to the communication server, use the **enable** command to enter your password if one is set, use the **configure** command to enter configuration mode, and add the following commands to your configuration:

```
appletalk service
interface ethernet 0
appletalk cable-range 0-0 0.0
! sets 500-CS into discovery mode
line 5 6
modem inout
speed 38400
arap enabled
autoselect
```

If you already know the cable-range and the zone names you need, include the information in the configuration file. If you do not know this information, let the communication server learn about the AppleTalk network in discovery mode by following these steps:

1. Permit the communication server to monitor the line for a few minutes.
2. Log in and enter configuration mode.
3. Show the configuration again (using the **show config** command).
4. Note the **appletalk cable-range** and **appletalk zone** variables.
5. Manually add the information in those two entries and add any user accounts.
6. Save the configuration.

```
appletalk cable-range 105-105 105.222
appletalk zone Marketing Lab
! Do not use quotation marks in this entry
username arauser password arapasswd
! Add as many users as you need
```

7 Show the configuration again (using the **show config** command) to make sure the configuration is correct.

Example of Setting up aTelebit T-3000 Modem

The following example describes how to set up a Telebit T-3000 modem that you are attaching to a 500-CS communication server, which supports hardware flow control. The Macintosh will use a CCL script to configure the attached modem.

Start with the modem at factory defaults. (AT&F9 is the preferred configuration for hardware flow control. Use the direct command if you have a terminal attached to the modem, or use the T/D Reset sequence described in the Telebit T-3000 manual to reset the modem to the &F9 defaults.)

Attach a hardware flow control-capable cable between the modem and the device with which you are configuring the modem. (At this point, the modem is in hardware flow control mode, with auto-baudrate-recognition, and can detect your speed between 300 and 38,400 bps at 8-N-1. However, the modem must receive the flow control signals from the device to which you have the modem attached.)

Send the modem the following commands:

```
ATS51=6 E0 Q1 S0=2 &D3 &R3 S58=2 &W
```

This sequence tells the modem to perform the following tasks:

- Lock your DTE interface speed to 38,400 bps.
- Turn “command echo” off.
- Do not send any result codes.
- Auto-answer on the second ring (Germany requires this, but elsewhere you can set it to answer on the first ring with “s0=1”.)
- When DTR is toggled, reset to the settings in NVRAM.
- CTS is always enabled if hardware flow control is disabled.
- Use full-duplex RTS/CTS flow control.
- Write these settings to NVRAM.

At this point, if you press the carriage return or type characters, no characters appear on your screen because the result codes are turned off. You can see if the modem is working by getting a list of its configuration registers using the following command:

```
AT&V
```

After the modem is configured, connect it to the communication server with a modem-to-RJ45 adapter (Cisco Part Number: CAB-5MODCM) and an RJ-45 cable to the lines(s) that you plan to use.

The following commands are compatible with the Telebit 3000 settings described in this section:

```
arap enable
autoselect
no escape-character
flowcontol hardware
modem ri-is-cd
speed 38400
```

If you are attaching a Telebit T-3000 modem to an ASM-CS communication server, use an RJ-11 adapter and a straight cable. For more information about attaching a Telebit T-3000 modem to an ASM-CS communication server, see the *ASM-CS Hardware Installation and Maintenance* publication.